

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2002-366029

(P2002-366029A)

(43) 公開日 平成14年12月20日 (2002. 12. 20)

(51) Int. Cl. <sup>7</sup>	識別記号	F I	テマコード* (参考)
G 0 9 C 1/00	6 1 0	G 0 9 C 1/00	6 1 0 B 5 J 1 0 4
	6 5 0		6 5 0 B

審査請求 未請求 請求項の数 6 O L (全 28 頁)

(21) 出願番号 特願2001-178407(P2001-178407)

(22) 出願日 平成13年 6 月13日 (2001. 6. 13)

(71) 出願人 000005223

富士通株式会社

神奈川県川崎市中原区上小田中 4 丁目 1 番  
1 号

(72) 発明者 伊藤 孝一

神奈川県川崎市中原区上小田中 4 丁目 1 番  
1 号 富士通株式会社内

(72) 発明者 武仲 正彦

神奈川県川崎市中原区上小田中 4 丁目 1 番  
1 号 富士通株式会社内

(74) 代理人 100062993

弁理士 田中 浩 (外 2 名)

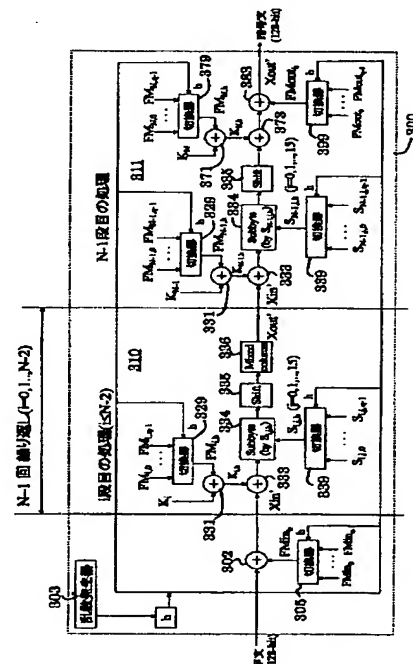
最終頁に続く

(54) 【発明の名称】 DPA に対して安全な暗号化

(57) 【要約】

【課題】 固定値を用いてマスクを行うことによって処理速度の向上と必要な RAM 領域の削減を実現する。

【解決手段】 暗号化装置 (300) は、乱数を発生する乱数発生器手段と、乱数に従って  $q$  個の固定値の中の 1 つを選択する第 1 の選択器 (329) と、乱数に従って  $q$  組の固定テーブルの中の 1 組を選択する選択器 (339) と、を具えている。排他的論理和手段 (333) は、固定値と鍵の排他的論理和と入力の排他的論理和をとる。非線形変換手段 (334) は、1 組の固定テーブルに従って非線形変換を行う。別の暗号化装置 (500) は、並列に結合された複数の暗号化部 (511、513) と、乱数に従ってその複数の暗号化部の中の 1 つを選択する選択器 (502) と、を具えている。



1

【特許請求の範囲】

【請求項 1】 排他的論理和手段および非線形変換手段を有する暗号化装置であって、乱数を発生する乱数発生手段と、 $q$  個 ( $q$  は整数) の固定値と、前記乱数に従って前記  $q$  個の固定値の中の 1 つを選択する第 1 の選択器と、を具え、前記排他的論理和手段は、前記選択された固定値と鍵の排他的論理和と前記排他的論理和手段の入力の排他的論理和をとるものであること、を特徴とする、暗号化装置。

【請求項 2】 排他的論理和手段および非線形変換手段を有する暗号化装置であって、乱数を発生する乱数発生手段と、 $q$  組 ( $q$  は整数) のマスクされた固定テーブルと、前記乱数に従って前記  $q$  組の固定テーブルの中の 1 組を選択する選択器と、を具え、前記非線形変換手段は、前記選択された 1 組の固定テーブルに従って入力を非線形変換するものであること、を特徴とする、暗号化装置。

【請求項 3】 乱数を発生する乱数発生手段と、並列に結合された複数の暗号化部と、前記乱数に従って前記複数の暗号化部の中の 1 つを選択する選択器と、を具え、前記複数の暗号化部の各々は、排他的論理和手段および非線形変換手段を含むものであること、を特徴とする、暗号化装置。

【請求項 4】 乱数を発生する乱数発生手段と複数の暗号化段とを具えた暗号化装置であって、前記複数の暗号化段の各々は、入力を非線形変換する非線形変換手段と、第 1 の入力と第 2 の入力の排他的論理和をとる排他的論理和手段と、を含み、前記排他的論理和手段の第 2 の入力は前記非線形変換手段の出力に結合されており、前記非線形変換手段は、 $q$  個 ( $q$  は整数) の固定値と、前記乱数に従って前記  $q$  個の固定値の中の 1 つを選択する選択器と、前記選択された固定値と鍵の排他的論理和と入力の排他的論理和をとる別の排他的論理和手段と、を有するものであること、を特徴とする、暗号化装置。

【請求項 5】 乱数を発生する乱数発生手段と、並列に結合された複数の暗号化部と、前記乱数に従って前記複数の暗号化部の中の 1 つを選択する選択器と、を具え、前記複数の暗号化部の各々は複数の暗号化段を有し、前記複数の暗号化段の各々は、入力を非線形変換する非線形変換手段と、第 1 の入力と第 2 の入力の排他的論理和をとる排他的論理和手段と、を含み、前記排他的論理和手段の第 2 の入力は前記非線形変換手段の出力に結合されているものであること、特徴とする、暗号化装置。

【請求項 6】 暗号化装置において用いられるプログラムであって、

乱数に従って複数の暗号化手順の中の 1 つを選択するス

2

テップと、前記選択された暗号化手順に従って、入力値を暗号化して出力するステップと、を実行させ、前記暗号化するステップは、固定値と鍵の排他的論理和と入力値の排他的論理和をとるステップと、1 組の固定テーブルに従って入力値を非線形変換するステップと、を含むものであること、を特徴とする、プログラム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、情報の暗号化の分野に関し、特に電力解析攻撃と呼ばれる暗号解読法に対する暗号のセキュリティ (安全) のための技術に関する。

【0002】

【発明の背景】暗号方式には一般的に公開鍵暗号方式と共通鍵暗号方式が含まれる。いわゆる共通鍵暗号方式は、暗号化と復号で同一の秘密鍵 (キー) を用いる。この秘密鍵を送信機側のユーザと受信機側のユーザの間で共有しその他の者に対して秘密にすることによって、暗号文が安全に送信できる。図 1 はスマートカードにおける共通秘密鍵を用いた暗号化の例を示している。図 1 において、スマートカードは、周知の形態で入力の明文 (プレーンテキスト) をその内部の暗号化処理部において共通秘密鍵を用いて暗号化して出力の暗号文を供給する。

【0003】暗号解読は、秘密鍵を含めた秘密情報を、暗号文等入手可能な情報から推定する。暗号解読の 1 つである電力解析攻撃が、Paul Kocher, Joshua Jaffe, and Benjamin Jun, "Differential Power Analysis" in proceedings of Advances in Cryptology-CRYPTO' 99, Springer-Verlag, 1999, pp. 388-397 に記載されている。この電力解析攻撃は、スマートカード等に搭載された暗号化プロセッサに様々な入力データを与えた時の電力消費データを収集および解析して、暗号プロセッサ内部の鍵情報を推定する。これは、公開鍵暗号と秘密鍵暗号の双方に適用できる。

【0004】電力解析攻撃には、単純電力解析 (以下、SPA という) (Single Power Analysis) および電力差分攻撃 (以下、DPA という) (Differential Power Analysis) が含まれる。SPA は暗号プロセッサにおける単一の電力消費データの特徴から秘密鍵の推定を行う。DPA は相異なる多数の電力消費データの差分を解析することによって秘密鍵の推定を行い、一般的には DPA の方が強力である。

【0005】例えば RSA 等の公開鍵暗号に対する DPA が、例えば、Thomas S. Messerges, Ezzy A. Dabbish and Robert H. Sloan "Power Analysis Attacks of Modular Exponentiation in Smartcards" Cryptographic Hardware and Embedded Systems (CHES' 99), Springer-Verlag, pp. 144-157 に記載されている。共通鍵暗号方式の現在の標準の DES (Data Encryptio

3

n Standards) に対するSPAおよびDPAが, Paul Kocher, Joshua Jaffe, and Benjamin Jun, "Differential Power Analysis," in proceedings of Advances in Cryptology-CRYPTO'99, Springer-Verlag, 1999, pp. 388-397に記載されている。共通鍵暗号方式の次世代標準となりうるラインデール(Rijndael)法に対するDPAが, 例えば, S. Chari, C. Jutla, J. R. Rao, P. Rohatgi, "An Cautionary Note Regarding Evaluation of AES Candidates on Smart-Cards," Second Advanced Encryption Standard Candidate Conference, March 1999に記載されている。

【0006】このように、電力解析攻撃の中でもDPAは特に有効な方法として注目されており、様々なDPA暗号解読法が研究されている。一方、DPA暗号解読を防止するための技術も開発されている。

【0007】DPAを適用可能な共通鍵暗号の通常の典型的な構成を説明する。図2、3および4は、典型的な共通鍵暗号で用いられる処理(operations)である鍵XOR(排他的論理和)、線形変換および非線形変換をそれぞれ示している。

【0008】図2において、鍵XORは入力データ $X_i$ の鍵情報 $K_i$ との排他的論理和(XOR)の出力 $Z_i$ を供給する。(但し、演算XORは図面および式では、 $\circ$ と $+$ を合成したシンボルで示されている。)図3において、線形変換 $L$ は入力データ $X_i$ に対する線形変換された出力 $Z_i = L(X_i)$ を供給する。ここで、任意の $x$ および $y$ について $L(x \text{ XOR } y) = L(x) \text{ XOR } L(y)$ である。線形変換には、ビット転置(permutation)および行列演算等がある。図4において、非線形変換 $W$ は、入力データ $X_i$ を非線形変換して出力 $Z_i = W(X_i)$ を供給する。ここで、任意の $x$ および $y$ について $W(x \text{ XOR } y) \neq W(x) \text{ XOR } W(y)$ である。非線形変換は、典型的には非線型変換テーブル $S_{box}$ をしばしば使い、入力 $X$ を $X = \{x_{u-1}, \dots, x_1, x_0\}$ ( $u$ は自然数)と $u$ 個に分割し、 $S_{box}$ である $w_i$ ( $i=0 \sim u$ )を用いて $z_i = w_i(x_i)$ を演算し、 $Z = (z_{u-1}, \dots, z_1, z_0)$ として結合

$$G_0(k') = \{G_i \mid z_i = w_i(m_i \oplus k'_i) \text{ の第 } e \text{ 桁のビット値} = 0 \} \quad (1)$$

【数2】

$$G_1(k') = \{G_i \mid z_i = w_i(m_i \oplus k'_i) \text{ の第 } e \text{ 桁のビット値} = 1 \} \quad (2)$$

ここで、 $e$ は、自然数であり、 $e$ 番目のLSBを表している。

【0013】次いで、仮定された $k_i'$ に対する次の電\*

$$DG(k') = (G_1 \text{ に属する電力消費曲線の平均}$$

$$- (G_0 \text{ に属する電力消費曲線の平均}) \quad (3)$$

【0014】図7(A)は $G_1$ に属する電力消費曲線の平均電力消費曲線を例示している。図7(B)は $G_0$ に属する電力消費曲線の平均電力消費曲線を例示している。仮定された鍵の要素値が真の鍵の要素値と等しく、

4

\*して出力 $Z$ を生成する。

【0009】典型的な共通鍵暗号では、これらの鍵XOR演算、線形変換および非線形変換の3つを適当に組み合わせて各ラウンド関数を構成し、そのようなラウンド関数を順に複数回(ラウンド)繰り返す。

【0010】DPAによる暗号解読技術を説明する。DPAは、電力消費データの測定の段階と、その電力消費データの差分に基づく鍵の解析の段階とを含んでいる。電力消費データの測定において、相異なる複数の符号のシーケンスを含んだ入力平文をスマートカード等の暗号化器にシリアルに供給して、その暗号化プロセッサによってその入力平文に応答して消費される電力消費の時間変化を例えばオシロスコープ等を用いて測定して、電力消費曲線を得る。図7(A)はそのような電力消費曲線を例示している。相異なる入力平文に対してその測定を行って統計的に充分な数の電力消費曲線を収集する。その測定で得られた複数の電力消費曲線の集合(セット)を $G$ とする。

【0011】次に電力消費曲線を用いた鍵の解析を説明する。図5は、従属接続の形の鍵XOR演算(図2)と非線形変換(図4)の組合せである暗号化の例を示している。この暗号化に対するDPAを説明する。図6は、図5における任意の非線形変換要素 $w_i$ に関する構成要素を取り出したものである。図6において、任意の入力平文における既知の数ビット値を $m_i$ とし、未知の鍵 $K$ における要素値を $k_i$ ( $K = \{k_{u-1}, \dots, k_1, k_0\}$ の中の1つの要素)とし、既知の $S_{box}$ テーブルにおける変換関数を $w_i$ とし、出力を $z_i (= w_i(m_i \text{ XOR } k_i))$ とする。DPAのために、プロセッサ内部で用いられている鍵の要素値を任意の $k_i'$ であると仮定する。既知の $m_i$ および $w_i$ とその仮定された $k_i'$ とから $z_i' = w_i(m_i \text{ XOR } k_i')$ を演算して、仮定された $k_i'$ に対する集合 $G(k_i')$ を次の部分集合 $G_0(k_i')$ および $G_1(k_i')$ に分ける。

【0012】

【数1】

\*力消費曲線の差分 $DG(k_i')$ を作成する。

【数3】

即ち $k_i' = k_i$ である場合には、図7(A)と(B)の差を表す図7(C)に例示された差分電力消費曲線中にスパイクが現われる。仮定された鍵の要素値が真の鍵の要素値と等しくなく、即ち $k_i' \neq k_i$ である場合に

5

は、図7(A)と(B)の差を表す図7(D)に例示された差分電力消費曲線は概ね平坦な曲線となる。従って、仮定した $k_i'$ に基づいて生成されたスパイクを有する差分電力消費曲線から、鍵 $k_i$ が推定できる。全ての $i$ に対する $k_i$ について差分電力消費曲線を生成することによって、最終的に鍵 $K$ を解読または確定(determine)することができる。

【0015】次に、 $k_i' = k_i$ である場合に、電力差分曲線 $DG(k_i')$ にスパイクが現れることを説明する。 $k_i' = k_i$ である場合、集合 $G(k_i')$ を式

(1) および式(2)に従って部分集合 $G_0(k_i')$  \*

$$(G_1 \text{に属する } z_i \text{ の平均HW}) - (G_0 \text{に属する } z_i \text{ の平均HW}) = 1 \quad (4)$$

【0016】一方、 $k_i' \neq k_i$ の場合は、仮定された $z_i' = w_i(m_i \text{ XOR } k_i')$ とそれに対応するプロセッサ内における実際のそれぞれの $z_i = w_i(m_i \text{ XOR } k_i)$ とが無関係なので、仮定された $z_i'$ に対する式(1)および式(2)に従って全ての $m_i$ に対する集合 $G(k_i')$ を部分集合 $G_0(k_i')$ および $G_*$

$$(G_1 \text{に属する } z_i \text{ の平均HW}) - (G_0 \text{に属する } z_i \text{ の平均HW}) = 0 \quad (5)$$

【0017】式(4)が成立する場合は $G_0(k_i')$ と $G_1(k_i')$ の間でのロード値 $z_i$ の平均ハミング・ウェイトに大きな差が生じるが、式(5)が成立する場合は $G_0(k_i')$ と $G_1(k_i')$ の間でのロード値 $z_i$ の平均ハミングウェイトに大きな差が生じない。

【0018】 $z_i = w_i(x_i)$ で示される変換 $w_i$ は、暗号化器内のROMやRAMなどのメモリ上から変換テーブル $S_{box}$ の出力値 $z_i$ がロード命令によって読み込まれることによって行われる。ロード命令が実行されるとき、ロード値のハミングウェイトに比例した電力が消費されるものと考えられている。その考えの妥当性を示す実験結果が、T. S. Messerge, Ezzy A. Dabbish and Robert H. Sloan, "Investigations of Power Attacks on Smartcards" Proceedings of USENIX Workshop on Smartcard Technology, Mar. 1999に記載されている。

【0019】従って、 $k_i' = k_i$ である場合、式

(4)が満たされることで消費電力の差が電力差分曲線上のスパイクという形になって現れるが、式(5)の場合はスパイクが現れず、概ね平坦な曲線となる。DPAは、図4に図3の線形変換 $L$ を挿入した暗号化器にも適用できることが判明している。

【0020】図8は、図4の暗号化器の前後に2つの線 \*

$$V(z) = a' + a_0 z_0 + a_1 z_1 + \dots + a_7 z_7 + a_{0,1} z_0 z_1 + a_{1,2} z_1 z_2 + \dots + a_{6,7} z_6 z_7 \quad (6)$$

【0022】これらの手法によれば、次の場合にDPAによって秘密鍵 $K$ が解読される。図9は、図4の暗号化器における電力消費曲線の測定対象点A、BおよびCを示している。

【0023】1. 入力 $M$ が既知かつ攻撃者が自由に選択可能即ち制御可能であり、鍵 $K$ が未知の固定値であり、

6

\*および $G_1(k_i')$ に分けると、仮定した $z_i' = w_i(m_i \text{ XOR } k_i')$ とこれに対応するプロセッサ内における実際の $z_i = w_i(m_i \text{ XOR } k_i)$ とが全ての $m_i$ に対して一致するので、 $z_i$ のハミング・ウェイトHW(Hamming weight)に関して次の式(4)が得られる。ハミング・ウェイトとは或る値をビット値で表現したときのビット値=1の個数である。例えば2進4ビット値(1101)<sub>2</sub>のハミングウェイトHWは3である。

【数4】

$*_1(k_i')$ に分けても、実際のそれぞれの $z_i$ (即ち $z_i'$ と仮定された実際の $z_i$ )に対しては2つの部分集合にランダムに分けたことになるので、次の式(5)が成立する。

【数5】

20 ★形変換を追加した構成を有する暗号化器を示している。 $L_1$ および $L_2$ をビット転置関数とし、 $w_i$ をDESの $S_{box}$ とすると、図8の構成はDESのF関数と等価になる。DESの仕様については、FIPS 46, "Data encryption standard" Federal Information Processing Standards Publication 46, U. S. Department of Commerce/ National Bureau of Standards, National Technical Information Service, Springfield, Virginia, 1977を参照されたい。図8の処理も、図6と同様の処理に変換でき、同様にDPAを適用して鍵 $K$ を推定できる。

【0021】以上の手法では、非線形変換中の $S_{box}$ 出力に注目してDPAを適用している。さらに、入力 $m_i$ と鍵 $k_i$ のXORの値(鍵XOR演算の出力)や $S_{box}$ への入力値 $x_i$ に対してDPAを適用する手法もある。また、或るプロセッサにおいては、隣接ビットモデルによって次の式(6)で表される電力消費がロード値のビットの関数で表現でき、それによってより有効な解析が可能になる場合がある。M. Akkar, R. Bevan, P. Discham p, and D. Moyart, "Power Analysis, What Is Now Possible..." Asiacrypt 2000において報告されている。

【数6】

$S_{box}$ の $w_i$ の変換が既知の場合。この場合、図9の測定対象点Aの所定タイミングにおける( $S_{box} w_i$ の出力の)電力消費曲線が測定される。

【0024】2. 入力 $M$ が既知かつ制御可能であり、鍵 $K$ が未知の固定値である場合。この場合、図9の測定対象点Bの所定タイミングにおける(鍵XOR演算の出力

7

の) 電力消費曲線が測定される。

【0025】3. 入力Mが既知かつ制御可能であり、鍵Kが未知の固定値である場合。この場合、図9の測定対象点Cの所定タイミングにおける(Sboxのw<sub>i</sub>をインデックスするためのロード入力)の電力消費曲線が測定される。

【0026】従来のDPA対策

従来のDPA対策法としては、例えばスマートカード内にノイズ発生器を置くことによって電力消費量の測定精度を落とす手法と、暗号アルゴリズムに対策を施す手法が存在する。その測定精度を落とす手法は容易に行えるが、測定回数を増加するなどの方法で解析が可能になるので根本的な対策とはならない。それに対して、暗号アルゴリズムへの対策は、容易でないが根本的な対策となりうる。暗号アルゴリズムに対策を施す手法の代表的なものとして、マスキング法と呼ばれているThomas S. Messerges, "Securing the AES Finalists Against Power Analysis Attacks," in proceedings of Fast Software Encryption Workshop 2000, Springer-Verlag, April 2000が知られている。マスキング法とは、入力値Mそのものを用いて暗号化の各処理を行う代わりに、乱数RをマスクとしてM' = M XOR Rで表される値M'を用いて暗号化を行う方法である。乱数Rは暗号化の各処理ごとに発生させるので、ここでは“乱数マスク値法”と呼ぶ。

【0027】次に、乱数マスク値法について説明する。図10は、乱数マスク値法による処理の概要図を示している。この処理は、図示のような上側の暗号化処理部と下側のマスク値生成部と乱数発生とで構成される。

【0028】図2、3および4に示されている通常の鍵XOR関数、線形関数および非線形関数が用いられる通常の暗号化処理を図10の暗号化処理に変更する場合は、それらは図10の暗号化において図11、12および13に示されているような乱数マスク値法による鍵XOR関数、線形関数および非線形関数に置換される。

【0029】乱数マスク値法では、暗号化における通常の間データX<sub>i</sub>を計算する代わりに排他的論理和X<sub>i</sub> = X<sub>i</sub>' XOR R<sub>i</sub>を満たすX<sub>i</sub>'および乱数R<sub>i</sub>が計算される。暗号化処理部においてX<sub>i</sub>'が計算され、マスク値生成処理部においてR<sub>i</sub>が計算される。図2と図11、図3と図12、および図4と図13に示されている処理(operation)におけるX<sub>i</sub>、X<sub>i</sub>'、Z<sub>i</sub>、Z<sub>i</sub>'、R<sub>i</sub>およびR<sub>O</sub><sub>i</sub>に関して、次の式(7)が成立する。

【数7】

$$\begin{cases} X_i = X_i' \oplus R_i \\ Z_i = Z_i' \oplus R_{O_i} \end{cases} \quad (7)$$

【0030】図2では入力値X<sub>i</sub>と鍵K<sub>i</sub>に対してZ<sub>i</sub> = X<sub>i</sub> XOR K<sub>i</sub>が演算されるのに対して、図11では

8

暗号化処理内部で乱数発生器によって乱数R<sub>K<sub>i</sub></sub>を発生させた後、入力値X<sub>i</sub>'と鍵K<sub>i</sub>に対してZ<sub>i</sub>' = X<sub>i</sub>' XOR K<sub>i</sub> XOR R<sub>K<sub>i</sub></sub>が演算される。また、マスク値生成においてR<sub>i</sub>とR<sub>K<sub>i</sub></sub>からR<sub>O<sub>i</sub></sub> = R<sub>i</sub> XOR R<sub>K<sub>i</sub></sub>が演算される。

【0031】図3ではZ<sub>i</sub> = L(X<sub>i</sub>)の線形変換が行われるのに対して、図12の暗号化処理ではZ<sub>i</sub>' = L(X<sub>i</sub>')の変換が行われ、マスク値生成ではR<sub>O<sub>i</sub></sub> = L(R<sub>i</sub>)の変換が行われる。

【0032】図4ではw<sub>1</sub>、w<sub>2</sub>、・・・、w<sub>u-1</sub>で表されるu個のSboxを用いた非線型変換が行われるのに対して、図13(A)の暗号化処理では、図のNewSboxを用いた処理によってRAM領域上にw<sub>i</sub>'<sub>1</sub>、w<sub>i</sub>'<sub>2</sub>、・・・、w<sub>i</sub>'<sub>u-1</sub>で表される新しいSboxが作成されて、これを用いて非線型変換が行われる。また、図13(A)のマスク値生成ではそのNewSboxを用いた処理が行われ、R<sub>i</sub>と内部で生成された乱数R<sub>O<sub>i</sub></sub>からw<sub>i</sub>'<sub>1</sub>、w<sub>i</sub>'<sub>2</sub>、・・・、w<sub>i</sub>'<sub>u-1</sub>が生成され、w<sub>i</sub>'<sub>1</sub>、w<sub>i</sub>'<sub>2</sub>、・・・、w<sub>i</sub>'<sub>u-1</sub>とR<sub>O<sub>i</sub></sub>が出力される。図13(B)はNewSboxの詳細構成を示している。NewSboxは、内部の乱数発生器を用いてR<sub>O<sub>i</sub></sub>を生成し、また、R<sub>i</sub> = r<sub>i</sub><sub>u-1</sub>・・・r<sub>i</sub><sub>1</sub>r<sub>i</sub><sub>0</sub>と、R<sub>O<sub>i</sub></sub> = r<sub>O</sub><sub>i</sub><sub>u-1</sub>・・・r<sub>O</sub><sub>i</sub><sub>1</sub>r<sub>O</sub><sub>i</sub><sub>0</sub>と、図12(B)内で用いられるSbox、w<sub>1</sub>、w<sub>2</sub>、・・・、w<sub>u-1</sub>とから、w<sub>i</sub>'<sub>j</sub>(x) = w(x XOR r<sub>i</sub><sub>j</sub>) XOR r<sub>O</sub><sub>i</sub><sub>j</sub>を満たすw<sub>i</sub>'<sub>j</sub>をj = 0、1、・・・、u-1について作成し、そしてそのR<sub>O<sub>i</sub></sub>とw<sub>i</sub>'<sub>j</sub>を出力する。

【0033】次に、乱数マスク値法の安全性について簡単に説明する。乱数マスク値法においては、図10および後で説明する図19中の各ラウンド(段)における図13(A)および(B)のSboxのw<sub>i</sub>'が乱数に従って変化するので、DPAにおいてSboxの内容を知ることができない。即ち、前述の条件1におけるSboxが既知であるという条件が成立しないので、図8の測定対象点Aの所定のタイミングにおいて測定された電力消費曲線を式(1)および(2)に従ってG<sub>0</sub>とG<sub>1</sub>に分けることができない。従って、乱数値マスク法を用いた暗号処理装置はDPAに対して安全である。同様に、前述の条件2および3における鍵XOR関数の出力における測定対象点BおよびSboxへの入力における測定対象点Cについても、測定毎に変化する乱数要素が加算されるので、鍵Kが固定であるという条件が成立しないので、DPAに対して安全である。

【0034】次に、乱数マスク値法を用いた暗号化の例として、ラインデール法を説明する。図14は、DPA対策のない通常のN段ラインデール処理の全体的構成を示している。N段ラインデール処理の各段は、XOR演算、サブバイト処理Subbyte(Substitute Byte

9

e)、シフト処理Shift、およびミクストコラム処理Mixedcolumnを含んでいる。但し、最後の段は、別のXOR演算を含んでおり、ミクストコラム処理を含んでいない。図14において、Nは、秘密鍵Ksecのビット数によって決定され、Ksecが128ビットの場合は $N=10$ とし、192ビットの場合は $N=12$ とし、および256ビットの場合は $N=14$ とする。Kiは拡大鍵(sub-key)と呼ばれる鍵である。図15は、ラインデル法の128/192/256ビット秘密鍵KsecからN+1個の128ビット拡大鍵K0、K1、・・・、KNを生成する拡大鍵生成器を示している。秘密鍵から拡大鍵を生成する方法は、<http://www.nist.gov/aes/>に示されているRijndaelの仕様書に記載されている。

【0035】図16はサブバイト処理Subbyteを示しており、この処理は、8ビット→8ビット変換SboxであるSを用いて、128ビット→128ビット非線形変換を行う。図17はシフト処理Shiftを示しており、この処理は、バイト単位の並べ替えを行う処理である。図18はミクストコラム処理Mixedcolumnを示しており、この処理は、行列を用いた体GF(28)上の演算を行う。

【0036】図19は、図14の通常のN段ラインデル法に対して、乱数マスク値法を適用したN段ラインデル法を示している。図19のN段ラインデル法は、図示のような上側のN段の暗号化処理部と下側のN段のマスク値生成部と乱数発生とで構成される。Kiはラインデル法の第i段の拡大鍵を表す。RKiはそれぞれの拡大鍵に対するランダムなマスク値を表す。サブバイト処理Subbyteは、図16に示されているような形態で16個のSbox: Si,0、Si,1、・・・、Si,15を用いて128ビット→128ビット非線形変換を行う。Si,0、Si,1、・・・、Si,15は第i段の新しいSbox“NewSbox”によって生成されたSboxである。図20は、NewSboxを示しており、入力値Riniから、内部で発生した乱数Routiに従って相異なる16個のSbox、Si,0(x)、Si,1(x)、・・・、Si,15(x)を生成し、乱数Routiを供給する。Shift処理およびMixedcolumn処理は、通常のラインデル法の処理と同様に、図17および図18に示す線形変換である。

【0037】図19の処理のフローをステップ[1101]～[1109]およびステップ[1201]～[1209]で示す。

[1101] i=0とセットする。

[1102] 乱数マスク値Rinを生成し、平文に対してRinとのXOR演算を行う(XORする)。

[1103] 平文に対してKiXORKiとのXOR演算を行う。マスク処理にマスク値RKiを入力して1

10

6個のSbox、Si,j(x)(j=0、1、・・・、15)を生成する。

[1104] ステップ[1103]で生成したSi,j(x)を用いたSubbyte処理を行う。

[1105] Shift処理およびMixedcolumn処理を行う。

[1106] i:=i+1とする。

[1107] i<N-1である場合はステップ[1103]に戻る。それ以外は次のステップに進む。

[1108] KN-1とのXOR処理を行い、RKN-1をマスク値処理部に供給して16個のSbox、SN-1,j(x)(j=0、1、・・・、15)を生成する。

[1109] SN-1,j(x)によるSubbyte処理、Shift処理、およびKNとのXOR演算を行う。

[1110] ステップ[1109]の演算出力に対して、マスク値生成の出力結果RoutとのXOR演算を行い、その結果の暗号文を出力として供給する。

【0038】マスク値生成処理のフロー:

[1201] i=0、Mask=Rinとセットする。但し、Rinは[1102]で生成された乱数マスク値である。

[1202] 暗号化処理部より受け取ったRKiから、MaskXORRKiを演算して、新しいMaskを生成する。

[1203] ステップ[1202]で生成した新しいMaskをNewSboxに入力することによって、16個のSbox、Si,j(x)(j=0、1、・・・、15)と乱数Routiを得て、Routiを新しいMaskとする。Si,j(x)は暗号化処理部の第i段のSubbyteにおいて用いられる。

[1204] MaskをShift処理およびMixedcolumn処理に供給し、その処理からの出力を新しいMaskとする。

[1205] i:=i+1とセットする。i<N-1である場合はステップ[1202]に戻る。

[1206] 暗号化処理部から入力されたRKN-1からMaskXORRKN-1を演算して、その演算結果を新しいMaskとする。

[1207] RinN-1をNewSboxに供給することによって、16個のSbox、SN-1,j(x)(j=0、1、・・・、15)と乱数RoutN-1を得て、RoutN-1を新しいMaskとする。SN-1,j(x)は暗号化処理部の第N-1段のSubbyteにおいて用いられる。

[1208] MaskをShift処理に供給し、その処理結果を新しいMaskとする。

[1209] 暗号化処理部より入力されたRKNから、MaskXORRKNを演算し、それを出力して処

理を終了する。

【0039】

【発明が解決しようとする課題】乱数マスク値法はDPAに対する安全性（セキュリティ）が高いことが知られているが、乱数マスク値法を採用した暗号化は、その暗号化の速度が通常の暗号化と比べて数分の1と低く、非常に大きいRAM領域を必要とするという欠点がある。

【0040】暗号化の速度が低い理由は、例えば暗号化処理におけるXOR演算では、通常の実装では2つの中間値xおよびyから $z = x \text{ XOR } y$ を演算するのに対して、乱数マスク値法では $x' = x \text{ XOR } r_x$ 、 $y' = y \text{ XOR } r_y$ を満たす中間値 $x'$ 、 $y'$ から $z' = x' \text{ XOR } y'$ を演算する他に、 $z'$ に関する新しいマスク値の生成処理 $r_z = r_x \text{ XOR } r_y$ を演算する必要があるからである。さらに、非線形変換において、通常Sboxという非線形変換テーブルをROMに保持するのに対して、乱数マスク値法では、新しいマスク値から非線形変換テーブルを毎回生成しなければならないので大量の計算が要求される。

【0041】大きいRAM領域を必要とする理由は、その非線形変換において、通常はその非線形変換テーブルをROMに保持するのに対して、乱数マスク値法では暗号化のたびに新しいSboxをRAMに格納するからである。例えば8ビット-8ビットの変換を行うSboxを用いるラインデール法において、DPA対策として乱数マスク値法を実装するには最低 $2^8 = 256$ バイト分のRAM領域が必要である。しかし、例えばST16（STマイクロエレクトロニクス社製）等の低コストのスマートカード用チップはRAM領域が128バイト程度しか持っていないので、乱数マスク値法を実装することは30 実際上不可能である。

【0042】乱数マスク値法の実装のために、マスク値を共有すること、および或る暗号化から次の暗号化までの間にマスク値を生成することなどによって、見かけの処理速度の向上および必要なRAM領域の削減等が考えられているが、最初に乱数値でマスクするので全体的な処理速度の向上および大幅な必要RAM領域の削減は不可能である。

【0043】発明者は、暗号化において、乱数値を用いてマスクするのではなくて、固定値を用いてマスクすることによって処理速度の向上と必要なRAM領域の削減を実現すると有利であると認識した。ここで、この固定値を用いてマスクを行う方法を、固定マスク値法と呼ぶ。

【0044】本発明の1つの目的は、共通鍵暗号を行う暗号プロセッサに対する効率的な暗号解読防止を実現することである。本発明の別の目的は、秘密鍵の推定を困難にし、暗号プロセッサの安全性を高めることである。

【0045】

【発明の概要】本発明の1つの特徴（側面）によれば、50

暗号化装置は排他的論理和手段および非線形変換手段を有する。その暗号化装置は、さらに、乱数を発生する乱数発生手段と、q個（qは整数）の固定値と、その乱数に従ってそのq個の固定値の中の1つを選択する第1の選択器と、を具えている。その排他的論理和手段は、その選択された固定値と鍵の排他的論理和とその排他的論理和手段の入力の排他的論理和をとる。

【0046】本発明の別の特徴によれば、暗号化装置は排他的論理和手段および非線形変換手段を有する。その暗号化装置は、さらに、乱数を発生する乱数発生手段と、q組（qは整数）のマスクされた固定テーブルと、その乱数に従ってそのq組の固定テーブルの中の1組を選択する選択器と、を具えている。その非線形変換手段は、その選択された1組の固定テーブルに従って入力を非線形変換する。

【0047】本発明のさらに別の特徴によれば、暗号化装置は、乱数を発生する乱数発生手段と、並列に結合された複数の暗号化部と、その乱数に従ってその複数の暗号化部の中の1つを選択する選択器と、を具えている。その複数の暗号化部の各々は、排他的論理和手段および非線形変換手段を含んでいる。

【0048】本発明のさらに別の特徴によれば、暗号化装置は、乱数を発生する乱数発生手段と複数の暗号化段とを具えている。その複数の暗号化段の各々は、入力を非線形変換する非線形変換手段と、第1の入力と第2の入力の排他的論理和をとる排他的論理和手段と、を含んでいる。その排他的論理和手段の第2の入力はその非線形変換手段の出力に結合されている。その非線形変換手段は、q個（qは整数）の固定値と、その乱数に従ってそのq個の固定値の中の1つを選択する選択器と、その選択された固定値と鍵の排他的論理和と入力の排他的論理和をとる別の排他的論理和手段と、を有する。

【0049】本発明のさらに別の特徴によれば、暗号化装置は、乱数を発生する乱数発生手段と、並列に結合された複数の暗号化部と、その乱数に従ってその複数の暗号化部の中の1つを選択する選択器と、を具えている。その複数の暗号化部の各々は複数の暗号化段を有する。その複数の暗号化段の各々は、入力を非線形変換する非線形変換手段と、第1の入力と第2の入力の排他的論理和をとる排他的論理和手段と、を含んでいる。その排他的論理和手段の第2の入力はその非線形変換手段の出力に結合されている。

【0050】本発明のさらに別の特徴によれば、暗号化装置において用いられるプログラムは、乱数に従って複数の暗号化手順の中の1つを選択するステップと、その選択された暗号化手順に従って、入力値を暗号化して出力するステップと、を含んでいる。その暗号化するステップは、固定値と鍵の排他的論理和と入力値の排他的論理和をとるステップと、1組の固定テーブルに従って入力値を非線形変換するステップと、を含んでいる。



13

【0051】本発明によれば、共通鍵暗号を行う暗号プロセッサに対する効率的なDPA対策を実現でき、DPAを用いた秘密鍵の解説(analyze)を困難にし、暗号プロセッサの安全性を高めることができる。

【0052】実施形態においては、通常は乱数値で入力や鍵をマスクする乱数マスク値法に対して、本発明の固定マスク値法に従って、複数個の固定値を用意し、それを乱数によって切替えることによって、乱数マスク値法と同様の効果を得る。固定マスク値法では、マスク値が特定の固定値に限定されるので、新しいマスク値を予め求めておくことによって処理速度を高くできる。また、予め固定マスク値の集合(セット)を用意し、それぞれの固定マスク値に対応した非線形変換テーブルをROMに用意することによって、小さいRAM領域を有するプラットフォーム上でも実装が可能になる。例えばST16等の低コストのスマートカード用のLSIチップでも6Kバイト程度の大きなROM領域を備えているので、固定値マスク法は低コストのスマートカードに適している。

【0053】

【発明の好ましい実施形態】次に、本発明の実施形態を説明する。実施形態において、暗号化における全てまたは一部の諸要素および諸処理は、集積回路等によってハードウェアの形態で実装してもよいし、またはプロセッサによって実行されるプログラムの形態で実装してもよい。

【0054】図21は本発明による第1のタイプの暗号化装置100の概略的構成を示している。図22および23は、暗号化装置100の鍵XOR演算および非線形変換の構成をそれぞれ示している。線形変換は図2に示されているものを用いればよい。

【0055】図21において、暗号化装置100は、乱数R(R=0, 1, ..., q-1)を発生する乱数発生器103と、その乱数に従って固定マスク値FM<sub>0, R</sub>の中の1つを選択して供給する切換部または選択器104と、入力の平文と選択された固定マスク値FM<sub>0, R</sub>との排他的論理和を演算する(XORする)(X<sub>in'</sub>=平文XORFM<sub>0, R</sub>)排他的論理和106と、入力X<sub>in'</sub>を受取り乱数Rに従って入力X<sub>in'</sub>を暗号化して出力X<sub>out'</sub>を生成する暗号化処理部101と、その乱数に従って固定マスク値FM<sub>N+1, R</sub>の中の1つを選択して供給する切換部105と、その出力X<sub>out'</sub>と選択された固定マスク値FM<sub>N+1, R</sub>との排他的論理和を演算して(XORして)暗号文(X<sub>out'</sub>XORFM<sub>N+1, R</sub>)を生成する排他的論理和107と、を含んでいる。ここで、Nは暗号化処理部101で使用される固定マスク値のセットの数を表す。

【0056】暗号化装置100は、さらに、作業メモリ用のRAM162と、固定マスク値と、固定非線形変換テーブルS<sub>box</sub>と、線形変換関数L等を格納するRO

14

M164と、ROM等のプログラム・メモリ160に格納されているプログラムに従ってこれらの諸処理要素103~107を制御するプロセッサ150とを含んでいる。代替構成として、プロセッサ150は、諸処理要素103~107に対応する機能を実装したメモリ160中のプログラムを実行することによって諸処理要素103~107を実現してもよい。この場合、図21はフロー図として見る事ができる。

【0057】暗号化処理部101において、図22の鍵XOR、図2の線形変換および図23の非線形変換の組み合わせによって規定されるラウンド関数が繰返し実行され、または縦続接続されたそのような複数のラウンド関数回路によってそのラウンド関数が実行される。

【0058】図22の鍵XORは、切換部109によって乱数Rに従って選択された固定マスク値FM<sub>i, R</sub>と鍵K<sub>i</sub>とをXORして或る値を供給し、その値と入力X<sub>i'</sub>とをXORして出力Z<sub>i'</sub>を供給する。図23の非線形変換は、切換部111~119によって乱数Rに従ってS<sub>box</sub>の要素w<sub>ij, R'</sub>(j=0, ..., u-1)を選択し、その選択された要素w<sub>ij, R'</sub>を用いて非線形変換する。

【0059】乱数マスク値法において示した式(7)の關係を用いると、図22の鍵XORにおいて図11のR<sub>i</sub>とR<sub>oi</sub>に対応して次の式(8)が得られる。

【数8】

$$RO_i = R_i \oplus FM_{i,R} \quad (8)$$

【0060】図23の非線形変換において用いられる図13(B)のNewS<sub>box</sub>のr<sub>i</sub>とr<sub>oi</sub>に対応して次の式(9)が得られる。

【数9】

$$w'_{i,R}(x'_i) = w_i(r_i \oplus x'_i) \oplus r_{oi} \quad (9)$$

【0061】図24は本発明による第2のタイプの暗号化装置200の概略的構成を示している。図25および26はその鍵XORおよび非線形変換の構成をそれぞれ示している。線形変換はいずれの暗号化処理部においても図2に示されているものを用いればよい。

【0062】図24において、暗号化装置200は、乱数Rを発生する乱数発生器203と、その乱数に従って固定マスク値FM<sub>0, R</sub>の中の1つを選択して供給する切換部204と、入力の平文と切換部204によって選択された固定マスク値FM<sub>0, R</sub>とをXORする(X<sub>in'</sub>=平文XORFM<sub>0, R</sub>)XOR(排他的論理和)206と、入力X<sub>in'</sub>を受取りその入力X<sub>in'</sub>を暗号化して出力X<sub>out'</sub>を生成する複数の暗号化処理部208~209と、入力X<sub>in'</sub>を受取り乱数Rに従って互いに並列に結合された暗号化処理部208~209の中の1つを選択してその選択された暗号化処理部に入力X<sub>in'</sub>を供給する切換部211と、同じ乱数Rに従って暗号化処理部208~209の中の同じ1つを選択してその選択された暗号化処理部からの出力X<sub>out'</sub>



を供給する切換器213と、その乱数に従って固定マスク値 $FM_{N+1, R}$ の中の一つを選択して供給する切換部205と、その出力 $X_{out}'$ と選択された固定マスク値 $FM_{N+1, R}$ とをXORして暗号文( $=X_{out}' \text{ XOR } FM_{N+1, R}$ )を生成するXOR207と、を含んでいる。ここで、Nは使用される固定マスク値の数を表す。図24において、2つの切換部211および213のうち的一方だけで選択してもよく、その他方の切換部を省いてもよい。

【0063】暗号化装置200は、さらに、作業メモリ用のRAM262と、固定マスク値、固定非線形変換テーブル $S_{box}$ と、線形変換関数 $L$ 等を格納するROM264と、ROM等のプログラム・メモリ260に格納されているプログラムに従ってこれらの諸処理要素203~211を制御するプロセッサ250とを含んでいる。代替構成として、プロセッサ250は、諸処理要素203~211に対応する機能を実装したメモリ260中のプログラムを実行することによって諸処理要素203~211を実現してもよい。この場合、図24はフロー図として見る事ができる。

【0064】暗号化処理部208~209は、固定マスク値 $FM_{i, j}$ および $S_{box}$ のセット $w_{i, j}$ が異なること以外は構成が共に同じである。各暗号化処理部208~209において、図25の鍵XOR演算、図2の線形変換処理および図26の非線形変換の組み合わせによって規定されるラウンド関数が繰返し実行され、または継続に結合されたそのような複数のラウンド関数回路によってそのラウンド関数が実行される。

【0065】図25の鍵XORは、暗号化処理部208~209の各々に固有の固定マスク値 $FM_{i, R}$ と鍵 $K_i$ をXORして或る値を供給し、その値と入力 $X_{i, R}'$ とをXORして出力 $Z_{i, R}'$ を供給する。図26の非線形変換は、その暗号化処理部に固有の $S_{box}$ のセット $w_{i, j, R}'$  ( $j=0, \dots, u-1$ )を用いて非線形変換する。ここで、乱数マスク値法における式(7)の関係をを用いると、図25の鍵XORにおいて図11の $R_i$ と $RO_i$ に対応して $RO_i = R_i \text{ XOR } FM_{i, R}$ が得られ、図26の非線形変換において図12(B)の $r_i$ と $ro_i$ に対応して $w_{i, j, R}'(x_{i, R}') = w_{i, j}(r_i \text{ XOR } x_{i, R}') \text{ XOR } ro_i$ が得られる。

【0066】本発明による上述の固定マスク値法は、低コストのスマートカードに実装するのに適した方法と考えられる。ここでは、次のような問題点を検討することによってより好ましい実装形態を実現する。

【0067】(A) DPA対策としての有効性の問題  
乱数マスク値法はDPAに対して安全であることが知られているが、固定マスク値法はDPAに対してどの程度安全であるかが不明であり、実装法および固定マスク値の条件しだいではDPAに対して弱い可能性がある。

【0068】(B) ROM領域が膨大になる可能性の問

## 題

固定マスク値法は、非線形変換テーブルをROMに用意することで、乱数マスク値法より必要なRAM領域を小さく抑えることができるという利点がある。しかし、用意されるマスク固定値の数に応じて使用ROM量が決まる。安全性を確保するために必要なROMの容量が膨大になる可能性がある。

【0069】(A)の問題に関連して、本発明を適用する或る暗号化では、或る条件の固定マスク値を使用すればその数 $q$ が2でも十分に安全であることが分かっている。本発明による暗号化法としてラインデール法を適用した場合の安全性を後で説明する。同様に(B)の問題に関連して、相異なる固定マスク値の数 $q$ を2程度にして、同じラウンドやラウンド毎の $S_{box}$ を共通にすることで、 $S_{box}$ のデータ量を、本発明が適用されない通常の実装の約2~数倍に抑えて、使用ROM量を小さく抑える。実施形態において暗号化法としてラインデール法を適用した場合の使用ROM量について後で述べる。

【0070】図27は、図21の第1のタイプの暗号化装置100の一例を示しており、ラインデール(Rijndael)暗号化法に固定マスク値法を適用した暗号化装置300を示している。但し、図27において、図21におけるプロセッサ150、メモリ160、162および164は簡単化のために省略されている。図27において、暗号化装置300は、乱数 $h$  ( $h=0, 1, \dots, q-1$ )を発生する乱数発生器303と、その乱数 $h$ に従って固定マスク値 $FM_{i, h}$ の中の一つを選択して供給する選択器305と、入力 $X_{in}$ と選択された固定マスク値 $FM_{i, h}$ とをXORするXOR302と、入力 $X_{in}$ を受取って乱数 $h$ に従って入力 $X_{in}$ を暗号化して出力 $X_{out}'$ を生成する複数段の暗号化処理段310 (各段 $i$ は $0 \leq i \leq N-2$ )と、第 $N-2$ 段の暗号化処理段310の出力 $X_{out}'$ を入力 $X_{in}$ として受取り乱数 $R$ に従ってその入力を暗号化して暗号文 $X_{out}'$ を生成する第 $N-1$ 段の暗号化処理段311と、を含んでいる。

【0071】複数段の暗号化処理段310の各段 $i$ は、乱数 $h$ に従って選択された固定マスク値 $FM_{i, h}$ を供給する選択器329と、鍵 $K_i$ と固定マスク値 $FM_{i, h}$ をXORして出力を供給するXOR331、その出力の値と入力 $X_{i, R}'$ とをXORするXOR333と、その乱数 $h$ に従って $S_{box}$ である固定値 $S_{i, j, h}$  ( $j=0, 1, \dots, 15$ )の中の一つを選択してマスクされた鍵 $K_{0, h}$ を供給する切換器または選択器339と、その選択された $S_{i, j, h}$ に従ってXOR333からの出力をサブバイトするSubbyte334と、Subbyte334の出力をシフトするShift335と、Shift335の出力をミクストコラムするMixedColumn336とを含んでいる。

17

【0072】第N-1段の暗号化処理段311は、第i段の暗号化処理段310と同様の切換器329および339と、XOR331および333と、Subbyte334と、Shift335とを有し、MixedColumnは有せず、切換器379によって選択された固定マスク値FMN<sub>h</sub>とKN<sub>h</sub>とをXOR371によってXORしてマスクされた鍵KN<sub>h</sub>を生成し、KN<sub>h</sub>とShift335の出力とをXOR373によってXORし、切換器399によって選択された固定マスク値FMout<sub>h</sub>とXOR373の出力とをXOR383によってXORして暗号文を生成する。

【0073】ここで、FMi<sub>h</sub>、FMi<sub>nh</sub>、およびFMout<sub>h</sub>は、固定マスク値である。Si<sub>j,h</sub>は、固定のSboxである。これらの固定マスク値およびSbox値は予め求められる。従って、固定マスク値法では通常の乱数マスク値法における逐次計算が不要なので、処理が高速化される。さらに、予め求めた固定マスク値およびSboxの変換テーブルを、例えば図21のRAM162および図24のRAM262のようなRAMにではなくて、図21のROM164および図24のROM264のようなROMに格納することによって、実装に必要なRAM領域を大幅に節約できる。このような必要なRAM量の節約は、ほんの128バイトのRAM領域しか持っていない低コストのスマートカードへの実装に有利である。

【0074】図27の暗号化装置300に入力平文が供給されると、内部の乱数発生器303によって0≤h≤q-1を満たす乱数hが生成される。その生成された乱数hに従って、図27に示す全ての切換器305、329、339、379および399において、q個の入力値の中からh番目のものが選択されて供給される。図28は図27におけるSubbyte334の構成を例示している。Shift335は図16に例示されているものを、Mixedcolumn336は図17に例示されているものを用いる。図28のSubbyteにおいて用いられるSbox Si<sub>j,h</sub>は、図16に示されている通常のラインデル法のSubbyteのSboxであるSを用いて次の式(10)で表される。

【数10】

$$S_{i,j,h}(x) = S(x \oplus a_{i,j,h}) \oplus b_{i,j,h} \quad (10)$$

【0075】式(10)におけるai<sub>j,h</sub>およびbi<sub>j,h</sub>は、式(9)のri<sub>j</sub>およびroi<sub>j</sub>にそれぞれ対応し、それぞれi番目のラウンドのSubbyteの入力および出力のマスク値に対応する。ai<sub>j,h</sub>は、入力マスク値なので、そのSubbyteより前のマスク値および処理等によって一意的に決まる。一方、bi<sub>j,h</sub>は任意に設定可能である。

【0076】図27の暗号化装置300処理のフローをステップ[1301]～[1314]に示す。ステップ

18

[1303]～[1309]は図28の第i段における処理である。ステップ[1310]～[1314]は図27の第N-1段における処理である。

[1301] i=0とセットする。

[1302] 入力平文を受け取り、次いで乱数発生器303によって乱数h(0≤h≤q-1)が発生される。この乱数hは後のステップで使用される。

[1303] 入力平文に対して、固定マスク値の集合{FMi<sub>0</sub>, ..., FMi<sub>q-1</sub>}の中からFMi<sub>nh</sub>を切換器305によって選択した後、XOR302によって入力平文とFMi<sub>nh</sub>をXORする。XOR302の出力を中間データXとする。

[1304] 固定マスク値の集合{FMi<sub>0</sub>, ..., FMi<sub>q-1</sub>}の中からFMi<sub>h</sub>を切換器329によって選択して、拡大鍵Ki、中間データXおよびFMi<sub>h</sub>に対してXXORKiXORFMi<sub>h</sub>を演算し、その演算結果を新しい中間データXとする。

[1305] Subbyte334において、中間データXに対して、切換器339によって乱数hに従って選択された非線形変換テーブルSi<sub>j,h</sub>(x)に従ってサブバイト処理する。その演算結果を新しい中間データXとする。

[1306] Shift335において中間データXをシフトする。そのシフトされたデータを新しい中間データXとする。

[1307] Mixedcolumn336において中間データXをミクストコラム処理する。その処理結果を新しい中間データXとする。

[1308] i:=i+1とセットする。

[1309] i<N-1の場合は[1303]に戻る。それ以外は次のステップに進む。

[1310] 乱数hに従って固定マスク値の集合{FMN-1,0, ..., FMN-1,q-1}の中からFMN-1,hを切換器329によって選択して、拡大鍵KN-1、中間データXおよびFMN-1,hに対して、XXORKN-1XORFMN-1,hを演算する。その演算結果を新しい中間データXとする。

[1311] 切換器339によって選択された非線形変換テーブルSN-1,j,h(x)に従ってサブバイト処理を行う。その処理結果を新しい中間データXとする。

[1312] Shift335において中間データXをシフトする。そのシフトされたデータを新しい中間データXとする。

[1313] 乱数hに従って固定マスク値の集合{FMN,0, ..., FMN,q-1}の中から切換器379によってFMN,hを選択して、その中間データX、拡大鍵KNおよびFMN,hに対して、XXORKNXORFMN,hを演算する。その演算結果を新しい中間データXとする。

19

【1314】 乱数 $h$ に従って固定マスク値の集合 $\{FMout_0, \dots, FMout_{q-1}\}$ の中から $FMout_h$ を切換器399によって選択して、中間データ $X$ と $FMout_h$ をXORし、その演算結果を出力暗号文 $Xout'$ として供給する。

【0077】図29は、ラインデール法に固定マスク値法を適用した別の暗号化装置400を例示している。但し、図29において、図21におけるプロセッサ150、メモリ160、162および164は簡単化のため\*

$$FM_{i,h} = \begin{cases} C_h \oplus FMin_h & (i=0) \\ C_h \oplus D_h & (i=1, \dots, N-1) \\ D_h & (i=N) \end{cases} \quad (11)$$

【0078】ここで、 $C_h$ および $D_h$ は、16バイトの定数であり、1バイト定数 $c_{h,j}$ および $d_{h,j}$  ( $j=0, 1, \dots, 15$ )をそれぞれ用いて、次の式\*

$$\begin{cases} C_h = c_{h,15} \dots c_{h,1} c_{h,0} \\ D_h = d_{h,15} \dots d_{h,1} d_{h,0} \end{cases} \quad (12)$$

【0079】図29の暗号化装置400において、サブバイト処理Subbyte334は、切換器339によって選択された16個のSbox、 $S_{0,h}, S_{1,h}, \dots, S_{15,h}$ を用いて、図30に示す非線形変換処理を行う。16個の $S_{j,h}$ はDPA対策なしの通常のラインデール法における $S$ であり、および式(12)における $c_{h,j}, d_{h,j}$ を用いて、 $S_{j,h}(x) = S(x \oplus c_{h,j}) \oplus d_{h,j}$ と表される。

【0080】図29の暗号化装置400の処理フローにおいては、図27の暗号化装置400の処理フローにおけるステップ【1305】および【1311】において、切換器339によって乱数 $h$ に従って16個のSbox、 $S_{0,h}, S_{1,h}, \dots, S_{15,h}$ の中から選択された非線形変換テーブル $S_{j,h}$ に従ってサブバイト処理する。

【0081】図27の暗号化装置300ではSubbyte334のテーブルが各ラウンドで異なるのに対して、図29の暗号化装置400では全てのラウンドで共通のテーブルが使用される。これが可能な理由を説明する。まず、通常のラインデール処理におけるSubbyte

$$\begin{cases} c_{h,15} = c_{h,14} = \dots = c_{h,0} \\ d_{h,15} = d_{h,14} = \dots = d_{h,0} \end{cases} \quad (13)$$

【0083】即ち、図29の暗号化装置400で用いられるSboxに必要な使用ROM量は、図27の暗号化装置300の場合の $1/16$ とすることができる。以上から、式(13)を満たす図29の暗号化装置400を用いることによって、Sboxに必要なROM領域を図27の暗号化装置300を用いた場合の $1/(16N)$ とすることができる。

20

\*に省略されている。図29の構成は、その各ラウンド関数における各Subbyte334に結合された切換器339に供給されるSboxが互いに同じである以外は、図27と同じであり、同じ構成要素については再び説明はしない。図30は図29におけるSubbyte334の構成を例示している。マスク値 $FM_{i,h}$ は、任意の $h=0, 1, \dots, q-1$ について次の式(11)を満たす。

【数11】

\* (12) のように記述される。

【数12】

★te (図16)への入力値を $X$ と表す。これに対し、図30のSubbyte処理の入力値は $X \oplus C_h$ と表すことができる。これは、マスク値の間に式(11)の関係が成立するからである。 $C_h$ は段数 $i$ とは関係ない定数なので、式(10)における $S_{i,j,h}(x) = S(x \oplus a_{i,j,h}) \oplus b_{i,j,h}$ に対して、 $a_{i,j,h}$ を段番号 $i$ に関係ない定数に設定できる。また、 $b_{i,j,h}$ は、任意の定数なので、 $i$ に関係ない定数に設定できる。よって、図30に示すように $S_{j,h}(x) = S(x \oplus c_{h,j}) \oplus d_{h,j}$ で表される段数 $i$ に無関係なSboxを用いたサブバイト処理が可能となる。従って、図29の暗号化装置400で用いられるSboxに必要な使用ROM量は、図27の暗号化装置300の場合の $1/N$ とすることができる。

【0082】さらに、図29の暗号化装置400において、マスク値 $C_h$ および $D_h$ に関する条件式(12)に、次の式(13)に示す条件を加えることによって、使用するSboxの数(種類の数)を相異なる16セットからほんの1セットに減らすことができる。

【数13】

【0084】また、図29の暗号化装置400では入力側および出力側において $FMin_h$ および $FMout_h$ をそれぞれXORしているが、この処理は安全性に寄与しないことが判明しているため、省略できる。さらに、マスク済みの鍵 $K_i \oplus FM_{i,h}$ の値を用いることによって、鍵 $K_i$ と固定マスク値 $FM_{i,h}$ とのXOR演算を省略できる。これらの省略によって、切換器によ

21

る処理が少し増加するだけなので、DPA対策のないラインデール法の場合とほぼ同じ計算量で固定マスク値法を適用したラインデール法が実現できる。

【0085】図31は、図24の第2のタイプの暗号化装置200の一例を示しており、ラインデール法に固定マスク値法を適用した別の暗号化装置500を例示している。但し、図31において、図24におけるプロセッサ250、メモリ260、262および264は簡単化のために省略されている。図31において、暗号化装置500は、乱数hを発生する乱数発生器503と、乱数hに従って切換えを行う切換え器502および504と、切換え器502および504によって乱数hに従って選択される互いに並列に結合された第0〜第q-1のq個の暗号処理部511〜513と、を含んでいる。

【0086】暗号処理部511〜513の各々は、入力Xin'を受け取って出力Xout'を生成する複数段の暗号化処理段530（段iは $0 \leq i \leq N-2$ ）と、前段の出力を入力として受取りその入力を暗号化して出力Xout'を生成する第N-1段の暗号化処理段と531、を含んでいる。第0段〜第N-2段の暗号化処理段530は、固定マスク値、XOR523、それぞれのSubbyte525、Shift526およびMixedcolumn527を有し、それぞれの固定マスク値および固定Sboxに従って暗号化を行う。第0段の暗号化処理段530の前にはXOR521が結合されている。第N-1段の暗号化処理段531は、固定マスク値、XOR523、528および529、Subbyte525およびShift526を有し、それぞれの固定マスク値および固定Sboxに従って暗号化を行う。図31では、鍵KiとFMj、hとがXORされたKi XOR FMj、hの値が直接供給されるように示されているが、図29の場合と同様に鍵KiとFMj、hとがXORによってXORされてXOR523〜553の入力に供給されるようにしてもよい。図31においては、図29の場合と同様にFMinhとFMouthとを省略できる。

【0087】図31の暗号化装置500においても、図29の暗号化装置400の場合と同様に全てのラウンドで同じSboxを使用しており、式(13)の条件を満た\*

$$c_{0,j} \oplus c_{1,j} = (10101010)_2 \text{ または } (01010101)_2 \quad (14)$$

$q \geq 2$ 、Sboxのセットの数が1、全ての $j = 0, 1, \dots, 15$ について ※【数15】

$$(c_{0,j} \oplus c_{1,j}) \vee (c_{1,j} \oplus c_{2,j}) \vee \dots \vee (c_{q-2,j} \oplus c_{q-1,j}) = (11111111)_2 \quad (15)$$

$q \geq 2$ 、Sbox数が1、全ての $j = 0, 1, \dots, 15$ について ★【数16】

$$(d_{0,j} \oplus d_{1,j}) \vee (d_{1,j} \oplus d_{2,j}) \vee \dots \vee (d_{q-2,j} \oplus d_{q-1,j}) = (11111111)_2 \quad (16)$$

ここで、( )<sub>2</sub>は2進法の値を表す。

【0092】DPAは、図9の測定対象点Aにおける所

22

\*たすようにすれば、図29の暗号化装置400の場合と同様のROMの削減が可能となる。暗号化装置500の計算量は、暗号化装置400の場合より切換え器が少ないだけ有利であり、DPA対策のないラインデール法とほぼ同じ計算量となる。但し、図31の暗号化装置500は、図29の暗号化装置400の場合よりもシフトおよびミクストコラム処理の数が増えるので、暗号化装置500の方が回路のサイズが大きくなる。なお、暗号化装置500には2つの選択器502および504が存在するが、左右の切換え器502および504のうち一方だけで切換えてもよい。そのような場合、他方の選択器は省略できる。

【0088】図29および図31の暗号処理装置においては同じ処理が異なる構成で行われるだけなので、安全性は共に同じである。

【0089】次に、固定マスク値法の安全性について説明する。図27の暗号化装置では、固定マスクの数qが充分大きい場合は、乱数マスク値法と実質的に同じ動作になるので、同様に高い安全性を有する。乱数マスク値法におけるほんの1ラウンドに簡略化された暗号関数について安全性が証明できるので、各ラウンドに同じSboxを使用する図29および31の暗号化装置400および500についても同様に安全性が証明できる。

【0090】次に、qが小さい場合の固定マスク値法の安全性を説明する。q=1では安全でないことが分かっている。次に小さいq=2の場合の安全性を評価する。図29および31の暗号化装置400および500において、q=2とし、式(13)に従ってSubbyteにおいて1セット（種類）だけのSboxを用い、かつFMinhおよびFMouthを省略した最も単純な場合においても、 $c_{0,j}, c_{1,j}, \dots, c_{q-2,j}$ に関して次の式(14)または式(15)の条件を設定し、 $d_{0,j}, d_{1,j}, \dots, d_{q-2,j}$ に関して次の式(16)の条件に設定することによって、DPAに対する安全性を高めることができる。

【0091】q=2、Sboxのセット（種類）の数が1、全ての $j = 0, 1, \dots, 15$ について 【数14】

定タイミングで行われる場合と、測定対象点BおよびCにおける所定タイミングで行われる場合とがある。図2

23

9 および31の暗号化装置400および500が安全であることを説明する。ビット(x, e)はxの第e桁のビット値を表す。

【0093】攻撃者は、鍵の推定のために以下の(i)および(ii)の処理を行う。

(i) DPAを用いて、チェックする仮の鍵(可能な鍵)の数を限定する。

(ii) (i)で限定された数の仮の鍵について、プロセッサ内部で用いられている真の鍵と値が一致するかどうかをチェックすることによって鍵を推定する。1つの仮の鍵すなわち1つのパターンの鍵をチェックするための計算の量を1単位(サイクル)とする。

【0094】上述の鍵の値のチェックは、暗号プロセッサの平文と暗号文の関係を調べることによって実現できる。即ち、暗号化プロセッサにより平文を暗号化した場合と、ソフトウェアなどの別の手段によって、そのチェックされる仮の鍵の値を用いてその平文を暗号化した場合とを比較する。平文と暗号文の関係が両方の暗号化で一致した場合は、その仮の鍵の値が暗号プロセッサ内部で用いられていると判定する。両者が一致しない場合は、その仮の鍵は用いられていないと判定する。

【0095】DPAを用いて128ビットの鍵を推定す\*

$$WD_j = (d_{0,j} \oplus d_{1,j}) \vee (d_{1,j} \oplus d_{2,j}) \vee \cdots \vee (d_{q-2,j} \oplus d_{q-1,j})$$

について、

$$\begin{cases} WD_j = (11111111)_2 \text{ならば、} f_j = 0, \\ \text{それ以外の場合は、} f_j = 1, \end{cases}$$

【0097】よって、j=0, 1, ..., 15の全てについてf<sub>j</sub>=0の場合に、暗号化装置はDPAに対して最も安全である。そのためには、全てのj=0, 1, ..., 15について、WD<sub>j</sub>=(11111111)<sub>2</sub>でなければならない。このとき、DPAを用いて鍵を求めるのに必要な計算量は最大の2<sup>128</sup>となる。

【0098】次に、鍵XORの出力(図9の測定対象点B)またはSboxへの入力(図9の測定対象点C)におけるDPAに対する安全性を説明する。図9の測定対象点BおよびCにおける所定タイミングにおけるDPAに対する安全性は、暗号プロセッサでRAMへのロードが行われた時の測定電圧とロード値との関係がどのようなモデルを用いて近似できるかによって異なる。まず、任意モデルの場合のDPAを考え、その後で式(6)で表される隣接ビット・モデルの場合のDPAを考える。

24

\*るのに必要な計算量の例を示す。例えば、DPAを行っても128ビット鍵に関する有益な情報を得られなかった場合は、128ビット鍵の全ての可能なパターンをチェックする必要があるので、必要な計算量は2<sup>128</sup>単位である。例えば、DPAによって128ビット鍵の最下位ビット(LSB)が0であることが分かった場合は、その鍵の残りの127上位ビット(MSBs)の全ての可能なパターンをチェックすればよいので、必要な計算量は2<sup>127</sup>単位である。

【0096】次に、Sboxの出力のDPAに対する安全性について、図9の測定対象点AにおけるDPAを図29および31の暗号化装置400および500に対して適用した場合について説明する。図29および31の第0段のSubbyteにおいて、それぞれのSboxの出力値がロードされるタイミングにおいてDPAを適用すると、鍵K<sub>i</sub>は28(16-F)=2<sup>128-8</sup>Fに比例した計算量で解読できることが分かっている。ここで、F=f<sub>0</sub>+f<sub>1</sub>+...+f<sub>15</sub>である。f<sub>j</sub>は、次のように定義される。j(j=0, 1, ..., 15)番目のSboxの出力マスク値をd<sub>0,j</sub>, d<sub>1,j</sub>, ..., d<sub>q-1,j</sub>としたときに、

【数17】

【0099】任意モデルの場合について、図9の測定対象点BおよびCにおける所定タイミングにおけるDPAを、式(13)によってサブバイト処理で用いられるSboxの数を1つ(1セット)だけとした図29および31の暗号化装置400および500に対して適用した場合について説明する。第0段のSubbyte処理において、それぞれの鍵XORの出力値がロードされるタイミングにおいてDPAを適用すれば、鍵K<sub>i</sub>は2<sup>128-(15/16)H</sup>に比例した計算量で解読できることが分かっている。ここで、H=h<sub>0</sub>+h<sub>1</sub>+...+h<sub>15</sub>である。h<sub>j</sub>は次のように定義される。j(j=0, 1, ..., 15)番目のSboxの入力マスク値をc<sub>0,j</sub>, c<sub>1,j</sub>, ..., c<sub>q-1,j</sub>としたときに、

【数18】

25

$$WC_j = (c_{0,j} \oplus c_{1,j}) \vee (c_{1,j} \oplus c_{2,j}) \vee \cdots \vee (c_{q-2,j} \oplus c_{q-1,j})$$

について、

$$WC_j = (wc_{j,7} wc_{j,6} wc_{j,5} wc_{j,4} wc_{j,3} wc_{j,2} wc_{j,1} wc_{j,0})_2$$

において、

$$h_j = (e=0, 1, \dots, 7 \text{ に関して } wc_{j,e} = 0 \text{ が成立する } e \text{ の個数})$$

【0100】即ち、 $j=0, 1, \dots, 15$  について  $h_j=0$  の場合がDPAに対して最も安全である。そのためには、全ての  $j=0, 1, \dots, 15$  について  $WC_j = (11111111)_2$  でなければならない。このとき、DPAを用いて鍵を求めるのに必要な計算量は最大の  $2^{128}$  になる。

【0101】次に、隣接ビット・モデルの場合について、 $q=2$  かつ式(13)によりサブバイト処理で用いるSboxのセット(種類)の数を1に限定し、式

(6)で表される隣接ビット・モデルを適用した場合について説明する。隣接ビット・モデルは、低コストのスマートカードにおける電圧を近似するのに適したモデル\*20

\*として知られており、このモデルが適用できる場合は、上述の任意モデルでは解読できない鍵情報を解読できる。図29および31の暗号化装置400および500における第0段のSubbyte処理において、それぞれのSboxの入力値がロードされるタイミング(図9の測定対象点Cにおける所定タイミング)にDPAを適用することによって、鍵 $K_i$ を  $128 - (15/16)H$  に比例した計算量で解読できる。ここで、 $H = h_0 + 1 + \dots + h_{15}$  である。

$h_j$  は、次のように定義される。 $j$  番目のSboxの入力マスク値を  $c_{0,j}$  および  $c_{1,j}$  としたときに、

【数19】

$$WC_j = c_{0,j} \oplus c_{1,j} = (wc_{j,7} wc_{j,6} \cdots wc_{j,0})_2 \text{ において、}$$

$$h_j = (e=0, 1, \dots, 7 \text{ に関して } wc_{j,e} = 0 \text{ を満たす } e \text{ の個数}) +$$

$$(e=0, 1, \dots, 7 \text{ に関して } wc_{j,e} = wc_{j,e+1} = 1 \text{ を満たす } e \text{ の個数})$$

ここで、 $wc_{j,e}=0$  または1である。 $j$  はSboxの序数(番号)を表す。 $e$  はビット位置を表す。

【0102】Sboxの数は1なので、条件式(14)が  $c_{0,j} \oplus c_{1,j} = (01010101)_2$  または  $(10101010)_2$  である場合は、 $H$  は最小値64となり、DPAを用いて鍵を求めるのに必要な計算量は最大でも  $2^{68}$  である。

【0103】以上のように、 $c_{0,j}, c_{1,j}, \dots, c_{q-1,j}$  に関して条件式(14)または式(15)を満たし、かつ  $d_{0,j}, d_{1,j}, \dots, d_{q-1,j}$  について式(16)の条件を満たすことによって、本発明による固定マスク値法を適用した暗号化装置に対してDPAを用いて128ビットのラインデル※

※法の秘密鍵を求める場合、表1および表2に示すように  $2^{128}$  または  $2^{68}$  に比例した計算量が必要となる。

但し、DPAに対する計算量についての安全性の閾値は  $2^{64}$  である。なお、計算量  $2^{68}$  は鍵の全てのビット・パターンを計算する場合の計算量  $2^{128}$  と比較すると小さいが、そのような鍵は実際の制限された時間内で解読することは実際上不可能である。従って、本発明による固定マスク値法を適用した暗号プロセッサに対してDPAを用いた場合も秘密鍵の解読は実際上不可能である。

【0104】

【表1】

表1. 固定マスク値法による、Sbox出力値ロードに対するDPAから128

ビット秘密鍵を求めるのに必要な計算量とマスク値の関係

WD <sub>j</sub> =(d <sub>0,j</sub> ⊕d <sub>1,j</sub> )∨(d <sub>1,j</sub> ⊕d <sub>2,j</sub> )∨...∨(d <sub>q-2,j</sub> ⊕d <sub>q-1,j</sub> )の値	鍵の解読のための計算量 (q≥2, 任意モデル)
全ての $0 \leq j \leq 15$ に関して WD <sub>j</sub> =(11111111) <sub>2</sub> (固定マスク値法)	$2^{128}$
一般の場合	$2^{128-6F}$ $F=f_0+f_1+\dots+f_{15}$ $f_i=0$ (if WD <sub>j</sub> =(11111111) <sub>2</sub> ) $f_i=1$ (otherwise)

【0105】

50 【表2】

表2. 固定マスク値法による、SubbyteにおけるSboxが1つの場合、  
Sbox入力値ロードに対するDPAから128ビット秘密鍵を求めるのに必要  
な計算量とマスク値の関係

$WC_j = (c_{0,j} \oplus c_{1,j}) \vee (c_{1,j} \oplus c_{2,j}) \vee \dots \vee (c_{q-2,j} \oplus c_{q-1,j})$ の値	鍵の解説のための 計算量 ( $q \geq 2$ , 任意モデル)	鍵の解説のための計 算量 ( $q=2$ , 隣接ビット モデル)
全ての $0 \leq j \leq 15$ に関して (11111111) <sub>2</sub> (固定マスク値法)	$2^{128}$	$2^{20}$
全ての $0 \leq j \leq 15$ に関して (01010101) <sub>2</sub> , (10101010) <sub>2</sub> (固定マスク値法)	$2^{68}$	$2^{68}$
$WC_j = (wc_{1,7}wc_{1,6}wc_{1,5}wc_{1,4}wc_{1,3}wc_{1,2}wc_{1,1}wc_{1,0})_2$ (一般の場合)	$2^{128 - (15/16)H}$ $H = h_0 + h_1 + \dots + h_{15}$ $h_j = (wc_{1,j} = 0 \text{ の数}) +$ ( $e=0,1,\dots,7$ )	$2^{128 - (15/16)H}$ $H = h_0 + h_1 + \dots + h_{15}$ $h_j = (wc_{1,j} = 0 \text{ の数}) +$ ( $wc_{1,e} = wc_{1,e+1} = 1 \text{ の数}) +$ ( $e=0,1,\dots,7$ )
全ての $0 \leq j \leq 15$ に関して (00000000) <sub>2</sub> (安全性が最低の場合)	$2^8$	$2^8$

【0106】従って、固定マスク値法の安全性は次の通りである。

1. 2以上のqについて、固定マスク値法は、Sbox出力値のロードに対するDPAに対して、式(16)の条件を満たせば、鍵の推定に計算量 $2^{128}$ が必要なので、安全である。この計算量は鍵の全ての可能なパターンをチェックする場合の計算量と同じである。

【0107】2. 2以上のqについて、式(13)の条件を満たす場合、固定マスク値法は、Sbox入力値ロードに対するDPAに対して、隣接ビット・モデルでない任意のモデルについて式(15)の条件を満たせば、鍵の推定に計算量 $2^{128}$ が必要なので、安全である。

【0108】3.  $q=2$ について、式(13)の条件を満たす場合、固定マスク値法は、Sbox入力値ロードに対するDPAに対して、隣接ビットモデルについて条件式(14)の $c_{0,j} \text{ XOR } c_{1,j} = (01010101)_2$ または $(10101010)_2$ を満たせば、  
40 鍵の推定に計算量 $2^{68}$ が必要なので、有限時間内での推定は実際上不可能であり、従って安全である。

【0109】4.  $q \geq 3$ について、必要なROM容量は増加するが、固定マスク値法は、 $q=2$ と同じ解析方法が適用できないため、上述の2の場合と同様に式(15)の条件を満たせば、隣接ビット・モデルと任意モデル鍵の双方とも鍵の推定に計算量 $2^{128}$ が必要なので、安全である。

【0110】5. 上述の3の場合において、条件式(14)はSboxのセットの数を1つに限定している(q 50

$=2$ なので出力値の数は2個)が、Sboxのセットの数を相異なるn個とするとROM容量はn倍となるが、計算量は $28(n-1) \times n^{16}$ 倍必要となる。

【0111】本発明による固定マスク値法は、上述のラインデル法のようなSPN型の暗号以外に、DESのようなフェイステル(Feistel)型の暗号にも適用できる。図32(A)は通常のDESによる暗号化の構成を示している。図32(B)は図32(A)におけるF関数のより詳細な構成を示している。図32(B)において、F関数は、線形変換EおよびPと、非線形変換テーブル $S_1 \sim S_8$ を有する非線形変換 $S_1 \sim S_8$ とを有する。

【0112】図33(A)は、図21の第1のタイプの暗号化装置100の一例を示しており、図29の暗号化装置400に類似した形態で固定マスク値法を図32のDESによる暗号化に適用した暗号化装置700を示している。図33(B)は図33(A)におけるF関数のより詳細な構成を示している。但し、図33(A)において、図21におけるプロセッサ150、メモリ160、162および164は簡単化のために省略されている。

【0113】暗号化装置700は、乱数hを発生する乱数発生器701と、その乱数hに従って固定マスク値 $FM_{inh}$ の中の1つを選択して供給する選択器702と、入力平文と選択された固定マスク値 $FM_{inh}$ とをXORするXOR712と、入力を受取って乱数hおよび拡大鍵 $K_i$ に従って入力を暗号化して出力を生成する複数



29

(例えば16段)のF関数暗号化処理段710~720と、その乱数hに従って固定マスク値F<sub>Mouth</sub>の中の1つを選択して供給する選択器704と、F関数暗号化処理段720の出力と選択された固定マスク値F<sub>Mouth</sub>とをXORして暗号文を生成するXOR714と、を含んでいる。F関数暗号化処理段710~720の各々は、前段のXOR出力を受け取って図33(B)に示されたF関数を実行し、その出力と前段の出力をXOR(722~723)によってXORして出力を供給する。

【0114】図33(B)のF関数は、乱数hに従って選択された固定マスク値F<sub>Mi, h</sub>を供給する選択器759と、拡大鍵K<sub>i</sub>と固定マスク値F<sub>Mi, h</sub>をXORして出力を供給するXOR762、その出力の値と線形変換Eによって線形変換された入力X<sub>i</sub>'とをXORするXOR763と、その乱数hに従ってサブバイト処理SubbyteS<sub>i, h</sub>中の1つを選択してXOR763の出力を供給する選択器752および756と、非線形テーブルSboxS<sub>i, h</sub>に従ってサブバイト処理するSubbyteS<sub>i, h</sub>と、その乱数hに従ってサブ

バイト処理S<sub>i, h</sub>中の1つを選択してその出力を供給する選択器754および757と、選択器754および757の出力を線形変換して出力Z<sub>i</sub>'を供給する線形変換Pとを含んでいる。

【0115】図21におけるプロセッサ150は、メモリ160に格納されているプログラムに従って図33の暗号化装置700の諸処理要素701~763等を制御する。代替構成として、プロセッサ150は、諸処理要素701~763等に対応する機能を実装したメモリ160中のプログラムを実行することによって諸処理要素701~763等を実現してもよい。この場合、図33はフロー図として見る事ができる。

【0116】図34(A)は、図24の第2のタイプの暗号化装置200の一例を示しており、図31の暗号化装置500に類似した形態で固定マスク値法を図32のDESによる暗号化に適用した暗号化装置800を示している。図34(B)は図34(A)におけるF関数のより詳細な構成を示している。但し、図34(A)において、図24におけるプロセッサ250、メモリ260、262および264は簡単化のために省略されている。

【0117】図34(A)において、暗号化装置800は、乱数hを発生する乱数発生器801と、乱数hに従って切換えを行う切換え器802および804と、切換え器802および804によって乱数hに従って選択される複数の暗号処理部820~830と、を含んでいる。

【0118】図24におけるプロセッサ250は、メモリ260に格納されているプログラムに従って図34の暗号化装置800の諸処理要素801~862等を制御する。代替構成として、プロセッサ150は、諸処理要

30

素801~862等に対応する機能を実装したメモリ160中のプログラムを実行することによって諸処理要素801~862等を実現してもよい。この場合、図34はフロー図として見る事ができる。

【0119】暗号処理部820~830の各々は、入力を受け取って出力を生成する複数段(例えば16段)のF関数暗号化処理段840~850を含んでいる。F関数暗号化処理段840~850の各々は、前段のXOR出力を受け取ってK<sub>i</sub>XORF<sub>Mi, h</sub>に従って図34

(B)に示されたF関数を実行し、そのF関数の出力と前段の出力をXOR(822~823)によってXORして出力を供給する。

【0120】図34(B)のF関数は、拡大鍵K<sub>i</sub>と固定マスク値F<sub>Mi, h</sub>のXOR値と線形変換Eによって線形変換された入力X<sub>i</sub>'とをXORするXOR862と、非線形テーブルSboxS<sub>i, h</sub>によるサブバイト処理SubbyteS<sub>i, h</sub>(i=1, 2, ..., 8)と、サブバイト処理S<sub>i, h</sub>の出力を線形変換して出力Z<sub>i</sub>'を供給する線形変換Pとを含んでいる。

【0121】図33および34において、ラインデール法の場合と同様に入力マスクF<sub>Min</sub>は省略できるが、ラインデール法と同様な手法ではF<sub>Mouth</sub>は省略できない。図35は、フェイステル型暗号化装置における複数の段の間のマスクの影響の伝播を示している。図35において太線はマスクされている経路を示している。F<sub>Mouth</sub>が省略できない理由は、図35に示されているように、フェイステル型の暗号では或る段でマスクされたデータ(A)が、次段(B)だけでなくそれ以降の段(C)へも影響するからである。

【0122】従って、フェイステル型暗号では、前段のマスク値を次段で完全にキャンセルできない。図36は、フェイステル型暗号化装置におけるマスク発生からマスク値キャンセルまでの経路を示している。図36において、太線はマスクされている経路を示している。フェイステル型暗号ではマスクを行ってからマスク値をキャンセルするまでには図36に示されているように4段以上必要であり、4段以上にわたってマスク値をキャンセルできる。このマスク値キャンセルの手法は、SPN型暗号化におけるように非線形変換の出力マスクが任意選択可能であることを利用するのでなくて、図36に示されているように前段からのマスクと等しい出力マスクを生成して、フェイステル型暗号化におけるXORによってキャンセルするものである。

【0123】この方法を用いれば暗号化処理最後のマスクF<sub>Mouth</sub>が不要になる。この手法をDES等の4ラウンド以上の暗号に適用する場合は、4ラウンドでキャンセルするような固定マスク値の構成を繰り返してもよいし、または全ラウンドの最後でキャンセルするような固定マスク値を用いてもよい。

【0124】上述の実施形態においては、全てのラウン

10

20

30

40

50

31

ドに対して固定マスク値法を適用する場合について説明した。しかし、DPAが可能となる条件で述べたように、DPAが成功するためには入力値が既知でかつ攻撃者がその値を制御可能でなければならない。従って、暗号化処理の最初の数ラウンドに対して固定値マスク法を適用すれば、その後のラウンドはその入力未知でかつ制御不可能なのでDPA対策は不要となる。それによって暗号化対策に必要な処理を削減できる。

【0125】表3は、ラインデル法の実装における固\*

32

\*定マスク値法を用いた暗号化の結果と、DPA対策なしの通常の暗号化、および従来の乱数マスク値法を用いた暗号化との比較を示している。表3において、Sは対策なしの場合の処理時間を表し、Rは対策なしの場合に必要なRAM量を表し、Mは対策なしの場合に必要なROM量を表す。但し、 $R \ll M$ である。安全性は攻撃者が鍵を推定するのに必要なチェックすべき可能な鍵の数で表される。

【表3】

表3. ラインデル法における、固定マスク値法適用、DPA対策なしの場合、および乱数マスク値法適用の比較 (但し  $R \ll M$ )

	対策なし	固定マスク値法		乱数マスク値法
		$q \geq 2$ 任意モデル	$q = 2$ 隣接モデル	
処理時間	S	$\approx S$	$\approx S$	$\gg S$ (3S~5S)
RAM量	R	$\approx R$	$\approx R$	$\geq R+M$
ROM量	M	$qM$	$2M$	$\approx M$
安全性	1	$\geq 2^{128}$	$2^{72}$	$\gg 2^{128}$

【0126】表4は、DESの実装における固定マスク値法を用いた暗号化の結果と、DPA対策なしの通常の暗号化、および従来の乱数マスク値法を用いた暗号化との比較を示している。

【表4】

表4. DESにおける、固定マスク値法適用、DPA対策なしの場合、および乱数マスク値法適用の比較 (但し  $R \ll M$ )

	対策なし	固定マスク値法	乱数マスク値法
		$q = 2$	
処理時間	S	$\approx S$	$\gg S$ (3S~5S)
RAM量	R	$\approx R$	$\geq R+M$
ROM量	8M	16M	$\approx 8M$
安全性	1	$\gg 2^{66}$	$\gg 2^{56}$

【0127】表3および表4より、乱数マスク値法では処理時間が長く大きいRAM量を必要とするのに対して、固定マスク値法では、2~3倍のROM量を必要とするが大きいRAM領域を必要とせず、対策なしの場合とほぼ同じ程度の処理時間で十分な安全性が確保できる。

【0128】上述の実施形態では主にラインデル法およびDESについて説明したが、固定マスク値法は、ラインデル法以外のSPN型暗号化、DES以外のフェイステル型暗号化、およびそれを組合せた型の暗号にも適用でき同様の有効性を示す。

【0129】以上説明した実施形態は典型例として挙げたに過ぎず、その変形およびバリエーションは当業者に

とって明らかであり、当業者であれば本発明の原理および請求の範囲に記載した発明の範囲を逸脱することなく上述の実施形態の種々の変形を行えることは明らかである。

【0130】(付記1) 排他的論理和手段および非線形変換手段を有する暗号化装置であって、乱数を発生する乱数発生器手段と、 $q$ 個の固定値(ここで $q$ は整数である)と、前記乱数に従って前記 $q$ 個の固定値の中の1つを選択する第1の選択器と、を具え、前記排他的論理和手段は、前記選択された固定値と鍵の排他的論理和と前記排他的論理和手段の入力の排他的論理和をとるものである、暗号化装置。

(付記2) さらに、 $q$ 組(セット)のマスクされた固

33

定テーブルと、前記乱数に従って前記q組の固定テーブルの中の1組を選択する第2の選択器と、を具え、前記非線形変換手段は前記選択された1組の固定テーブルに従って入力を非線形変換するものである、付記1に記載の暗号化装置。

(付記3) 前記第1の排他的論理和手段および前記非線形変換手段を有する暗号化部と、前記暗号化装置への入力と前記乱数に従って選択された或る固定値の排他的論理和をとる第2の排他的論理和手段と、前記乱数に従って選択された或る固定値と前記暗号化部の出力の排他的論理和をとる第3の排他的論理和手段とを具える、付記1に記載の暗号化装置。

(付記4) 排他的論理和手段および非線形変換手段を有する暗号化装置であって、乱数を発生する乱数発生器手段と、q組のマスクされた固定テーブル(ここでqは整数である)と、前記乱数に従って前記q組の固定テーブルの中の1組を選択する選択器と、を具え、前記非線形変換手段は、前記選択された1組の固定テーブルに従って入力を非線形変換するものである、暗号化装置。

(付記5) 複数の暗号化段を具え、この複数の暗号化段の各々は、その段の、前記排他的論理和手段と、前記固定テーブルと、前記選択器と、を有するものであり、前記複数の暗号化段のそれぞれの前記固定テーブルは同じものである、付記4に記載の暗号化装置。

(付記6) マスクされる前の固定テーブルを $S[x]$ とし、マスクされたj番目のテーブルを $S_j[x \text{ XOR } c_{i,j}] \text{ XOR } d_{i,j}$  ( $j=0, 1, \dots, 15$ )としたときに、 $(c_{0,j} \text{ XOR } c_{1,j}) \vee \text{ XOR } (c_{1,j} \text{ XOR } c_{2,j}) \vee \dots \vee (c_{q-2,j} \text{ XOR } c_{q-1,j}) = (11111111111111)_2$ である、付記4に記載の暗号化装置。

(付記7) テーブルの組の数 $q=2$ とする場合に、マスクされる前の固定テーブルを $S[x]$ とし、マスクされたj番目のテーブルを $S_j[x \text{ XOR } c_{i,j}] \text{ XOR } d_{i,j}$  ( $j=0, 1, \dots, 15$ )としたときに、 $c_{0,j} \text{ XOR } c_{1,j} = (1011010101010101)_2$ または $(0110101010101010)_2$ である、付記4に記載の暗号化装置。

(付記8) マスクされる前の固定テーブルを $S[x]$ とし、マスクされたj番目のテーブルを $S_j[x \text{ XOR } c_{i,j}] \text{ XOR } d_{i,j}$  ( $j=0, 1, \dots, 15$ )としたときに、 $(d_{0,j} \text{ XOR } d_{1,j}) \vee (d_{1,j} \text{ XOR } d_{2,j}) \vee \dots \vee (d_{q-2,j} \text{ XOR } d_{q-1,j}) = (11111111111111)_2$ である、付記4に記載の暗号化装置。

(付記9) 前記非線形変換手段はサブバイト手段であり、さらに、入力をシフトする手段と、入力をミクストコラムする手段とを具える、付記4に記載の暗号化装置。

(付記10) 乱数を発生する乱数発生器手段と、並列

34

に結合された複数の暗号化部と、前記乱数に従って前記複数の暗号化部の中の1つを選択する選択器と、を具え、前記複数の暗号化部の各々は、排他的論理和手段および非線形変換手段を含むものである、暗号化装置。

(付記11) 前記選択された暗号化部の前記排他的論理和手段は、固定値と鍵の排他的論理和と前記排他的論理和手段の入力の排他的論理和をとるものである、付記10に記載の暗号化装置。

(付記12) 前記非線形変換手段は、固定テーブルに従って前記非線形変換手段の入力を非線形変換するものである、付記10に記載の暗号化装置。

(付記13) 前記複数の暗号化部の各々は、その暗号化部への入力と固定値の排他的論理和をとる第2の排他的論理和手段と、入力と固定値の排他的論理和をその暗号化部の出力として生成する第3の排他的論理和手段と、を具えるものである、付記10に記載の暗号化装置。

(付記14) 前記選択された暗号化部の前記非線形変換手段は、固定テーブルに従って前記非線形変換手段の入力を非線形変換するものである、付記10に記載の暗号化装置。

(付記15) 前記複数の暗号化部の各々は複数の暗号化段を含んでおり、前記複数の暗号化段の各々は、その段の、鍵と固定値の排他的論理和と入力の排他的論理和をとる排他的論理和手段と、固定テーブルに従って入力を非線形変換する非線形変換手段とを含むものである、付記10に記載の暗号化装置。

(付記16) 乱数を発生する乱数発生器手段と複数の暗号化段とを具えた暗号化装置であって、前記複数の暗号化段の各々は、入力を非線形変換する非線形変換手段と、第1の入力と第2の入力の排他的論理和をとる排他的論理和手段と、を含み、前記排他的論理和手段の第2の入力は前記非線形変換手段の出力に結合されており、前記非線形変換手段は、q個の固定値(ここでqは整数である)と、前記乱数に従って前記q個の固定値の中の1つを選択する選択器と、前記選択された固定値と鍵の排他的論理和と入力の排他的論理和をとる別の排他的論理和手段と、を有するものである、暗号化装置。

(付記17) 前記非線形変換手段は、さらに、前記非線形変換手段の内部に、固定テーブルに従って入力を非線形変換する複数の非線形変換手段と、前記複数の非線形変換手段の中の1つを選択する選択器とを有するものである、付記16に記載の暗号化装置。

(付記18) 前記複数の暗号化段のそれぞれの前記非線形変換手段の固定テーブルは同じである、付記17に記載の暗号化装置。

(付記19) 前記複数の暗号化段における連続する複数の段においてマスクをキャンセルする、付記16に記載の暗号化装置。

(付記20) 前記複数の暗号化段において前記複数の

暗号化段の数より少ない数の暗号化段においてマスクが行われる、付記 16 に記載の暗号化装置。

(付記 2 1) 乱数を発生する乱数発生器手段と複数の暗号化段とを具えた暗号化装置であって、前記複数の暗号化段の各々は、入力を非線形変換する非線形変換手段と、第 1 の入力と第 2 の入力の排他的論理和をとる排他的論理和手段と、を含み、前記排他的論理和手段の第 2 の入力は前記非線形変換手段の出力に結合されており、前記非線形変換手段は、内部に、前記乱数に従って、固定テーブルに従って入力を非線形変換する非線形変換手段を有するものである、暗号化装置。

(付記 2 2) 乱数を発生する乱数発生器手段と、並列に結合された複数の暗号化部と、前記乱数に従って前記複数の暗号化部の中の 1 つを選択する選択器と、を具え、前記複数の暗号化部の各々は複数の暗号化段を有し、前記複数の暗号化段の各々は、入力を非線形変換する非線形変換手段と、第 1 の入力と第 2 の入力の排他的論理和をとる排他的論理和手段と、を含み、前記排他的論理和手段の第 2 の入力は前記非線形変換手段の出力に結合されているものである、暗号化装置。

(付記 2 3) 前記非線形変換手段は、固定値と鍵の排他的論理和と入力の排他的論理和をとる別の排他的論理和手段と、前記非線形変換手段の内部に、固定テーブルに従って入力を非線形変換する非線形変換手段と、を有するものである、付記 2 2 に記載の暗号化装置。

(付記 2 4) マスクされる前の固定テーブルを  $S_j[x]$  とし、マスクされた  $j$  番目のテーブルを  $S_j'[x \oplus c_{i,j}] \oplus d_{i,j}$  ( $j=0, 1, \dots, 7$ ) としたときに、 $(d_{0,j} \oplus d_{1,j}) \vee (d_{1,j} \oplus d_{2,j}) \vee \dots \vee (d_{q-2,j} \oplus d_{q-1,j}) = (1111)_2$  である、付記 2 2 に記載の暗号化装置。

(付記 2 5) 暗号化装置において用いるための記憶媒体に格納されたプログラムであって、乱数に従って  $q$  個の固定値の中の 1 つを選択するステップ（ここで  $q$  は整数である）と、前記選択された固定値と鍵の排他的論理和と入力値の排他的論理和をとるステップと、前記乱数に従って  $q$  組のマスクされた固定テーブルの中の 1 組を選択するステップと、前記選択された 1 組の固定テーブルに従って入力値を非線形変換するステップと、を実行させる、プログラム。

(付記 2 6) 暗号化装置において用いるための記憶媒体に格納されたプログラムであって、乱数に従って複数の暗号化手順の中の 1 つを選択するステップと、前記選択された暗号化手順に従って、入力値を暗号化して出力するステップと、を実行させ、前記暗号化するステップは、固定値と鍵の排他的論理和と入力値の排他的論理和をとるステップと、1 組の固定テーブルに従って入力値を非線形変換するステップと、を含むものである、プログラム。

(付記 2 7) 暗号化装置において用いるための記憶媒体に格納されたプログラムであって、入力値を非線形変換して出力を供給するステップと、第 1 の入力値と前記出力である第 2 の入力値の排他的論理和をとるステップと、を実行させ、前記非線形変換するステップは、乱数に従って  $q$  個の固定値の中の 1 つを選択するステップ

（ここで  $q$  は整数である）と、前記選択された固定値と鍵の排他的論理和と入力値の排他的論理和をとるステップと、前記乱数に関連付けられた固定テーブルに従って入力値を非線形変換するステップと、を含むものである、プログラム。

(付記 2 8) 暗号化装置において用いるための記憶媒体に格納されたプログラムであって、乱数に従って複数の暗号化手順の中の 1 つを選択するステップと、前記選択された暗号化手順に従って、入力値を暗号化して出力を供給するステップと、を実行させ、前記暗号化するステップは、入力値を非線形変換して出力を供給するステップと、第 1 の入力値と前記出力である第 2 の入力値の排他的論理和をとるステップと、を含むものである、プログラム。

【0131】

【発明の効果】本発明は、上述の特徴によって、共通鍵暗号を行う暗号プロセッサに対する効率的な暗号解読防止を実現でき、秘密鍵の推定を困難にし、暗号プロセッサの安全性を高めることができるという効果を奏する。

【図面の簡単な説明】

【図 1】図 1 は、図 1 はスマートカードにおける共通秘密鍵を用いた暗号化の例を示している。

【図 2】図 2 は、典型的な共通鍵暗号で用いられる鍵 XOR を示している。

【図 3】図 3 は、典型的な共通鍵暗号で用いられる線形変換を示している。

【図 4】図 4 は、典型的な共通鍵暗号で用いられる非線形変換を示している。

【図 5】図 5 は、縦続接続の形の鍵 XOR 演算（図 2）と非線形変換（図 4）の組合せである暗号化の例を示している。

【図 6】図 6 は、図 5 における任意の非線形変換要素  $w_i$  に関する構成要素を示している。

【図 7】図 7 (A) および (B) は、暗号化プロセッサがその入力平文に応答して消費する電力消費の時間変化を表す電力消費曲線を示している。図 7 (C) はスパイクを有する電力消費曲線の差分を示している。図 7

(D) はスパイクのない電力消費曲線の差分を示している。

【図 8】図 8 は、図 4 の暗号化器の前後に 2 つの線形変換を追加した構成を有する暗号化器を示している。

【図 9】図 9 は、図 4 の暗号化器における電力消費曲線の測定対象点 A、B および C を示している。

【図 10】図 10 は、乱数マスク値法による処理の概要

37

図を示している。

【図11】図11は、乱数マスク値法による鍵XORを示している。

【図12】図12は、乱数マスク値法による線形関数を示している。

【図13】図13は、乱数マスク値法による非線形関数を示している。

【図14】図14は、DPA対策のない通常のN段ラインデール処理の全体的構成を示している。

【図15】図15は、ラインデール法の秘密鍵 $K_{sec}$ から拡大鍵 $K_0, K_1, \dots, K_N$ を生成する拡大鍵生成器を示している。

【図16】図16は、サブバイト処理Subbyteの構成を示している。

【図17】図17は、シフトShiftの構成を示している。

【図18】図18は、ミクストコラム処理Mixedcolumnの構成を示している。

【図19】図19は、図14の通常のN段ラインデール法に対して、乱数マスク値法を適用したN段ラインデール法を示している。

【図20】図20は、図19の16個のSboxで使用するNewSboxを示している。

【図21】図21は、本発明による第1のタイプの暗号化装置の概略的構成を示している。

【図22】図22は、図21における鍵XORの構成を示している。

【図23】図23は、図21における非線形変換の構成を示している。

【図24】図24は、本発明による第2のタイプの暗号化装置の概略的構成を示している。

【図25】図25は、図24における鍵XORの構成を示している。

【図26】図26は、図24における非線形変換の構成を示している。

【図27】図27は、図21の第1のタイプの暗号化装置の一例を示している。

【図28】図28は、図27におけるSubbyteの構成を示している。

\*

38

\*【図29】図29は、図21の第1のタイプの暗号化装置の別の例を示している。

【図30】図30は、図29におけるSubbyteの構成を示している。

【図31】図31は、図24の第2のタイプの暗号化装置の一例を示している。

【図32】図32は、通常のDESの構成を示している。

【図33】図33は、図29の固定マスク値法をフェイステル型のDESに適用した構成を示している。

【図34】図34は、図31の固定マスク値法をフェイステル型のDESに適用した構成を示している。

【図35】図35は、フェイステル型暗号化装置における暗号における段の間のマスクの伝播を示している。

【図36】図36は、フェイステル型暗号化装置におけるマスク発生からマスク値キャンセルまでの経路を示している。

【符号の説明】

100 暗号化措置

101 暗号化処理部

103 乱数発生器

104、105 切換器

106、107 論理XOR

200 暗号化装置

203 乱数発生器

208～209 暗号化処理部

204、205 論理XOR

211、213 切換器

300 暗号化装置

310 第i段の処理

311 第N-1段の処理

303 乱数発生器

305、329、339、379、399 切換器

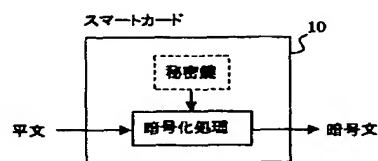
302、331、333、371、373、383 XOR (排他的論理和)

334 Subbyte

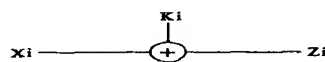
335 Shift

336 Mixedcolumn

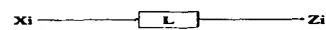
【図1】



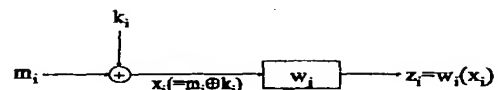
【図2】



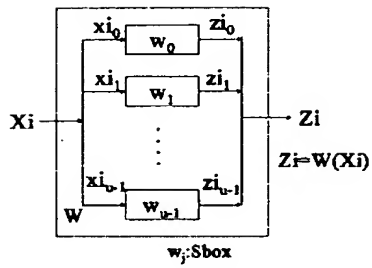
【図3】



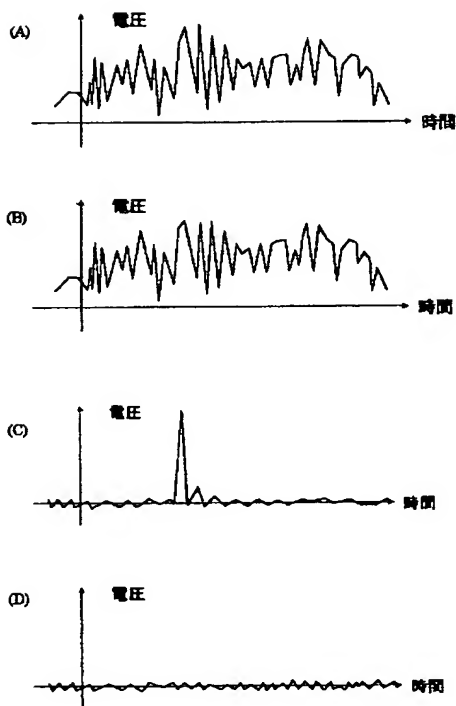
【図6】



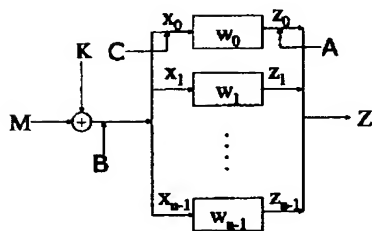
【図4】



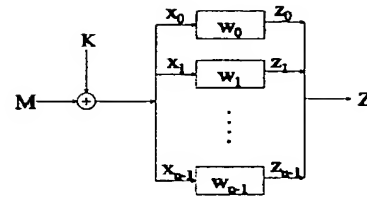
【図7】



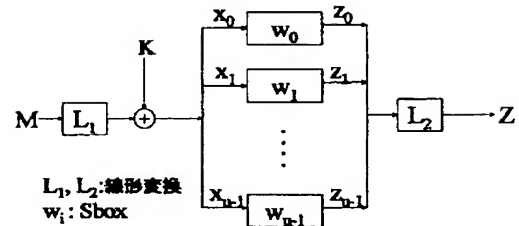
【図9】



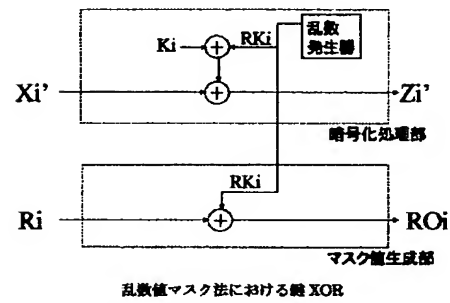
【図5】



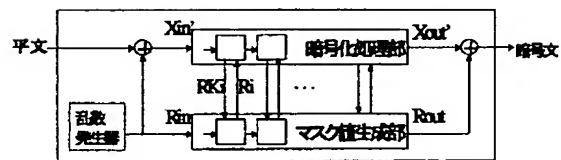
【図8】



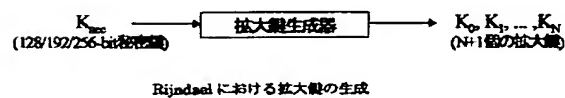
【図11】



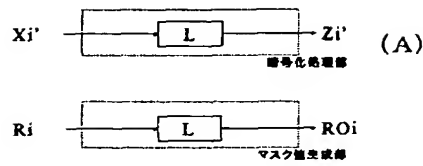
【図10】



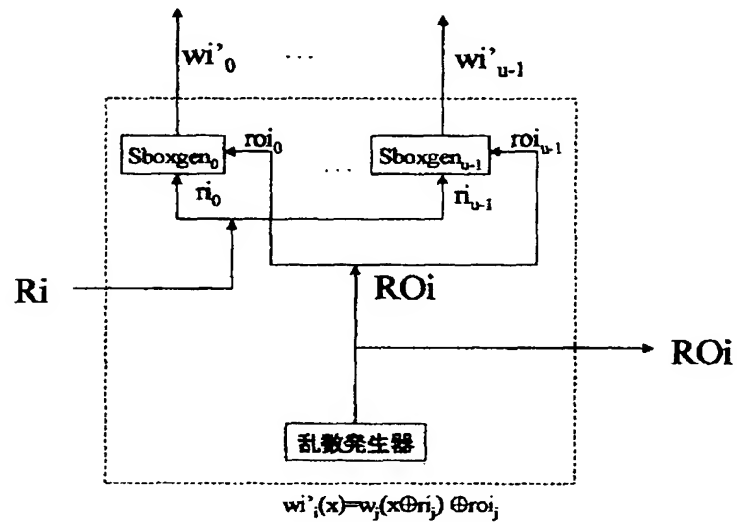
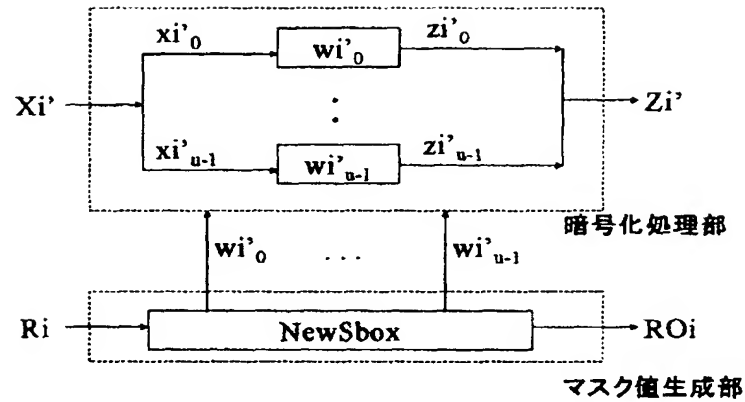
【図15】



【図12】

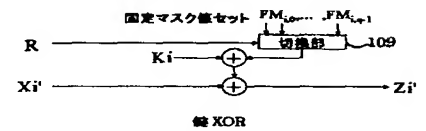
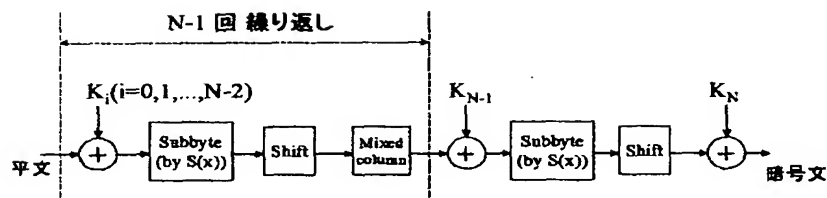


【図13】

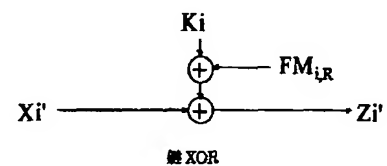


【図14】

【図22】

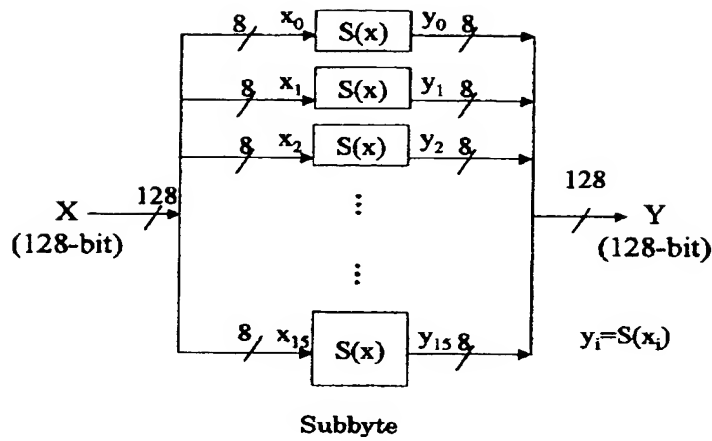


【図25】

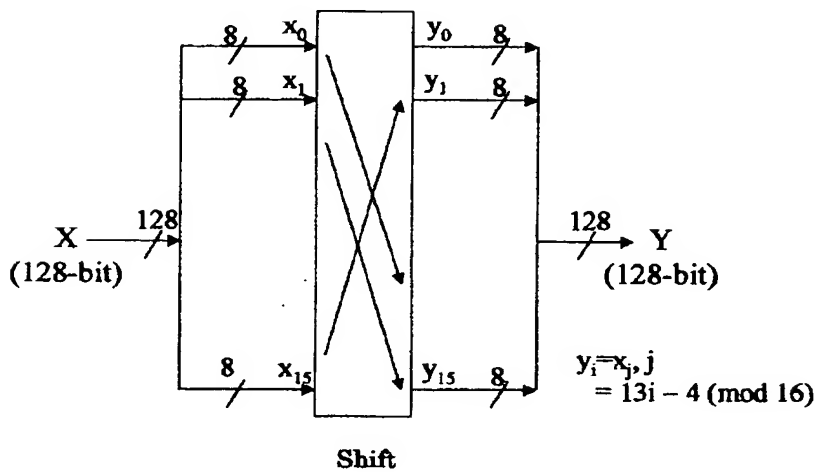




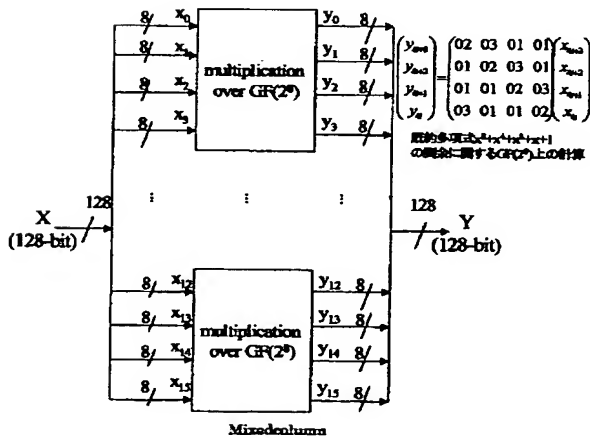
【図16】



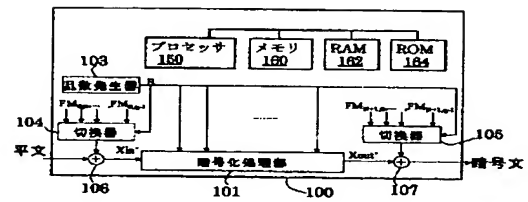
【図17】



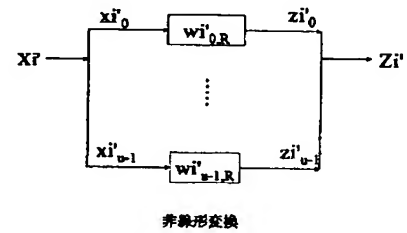
【図18】



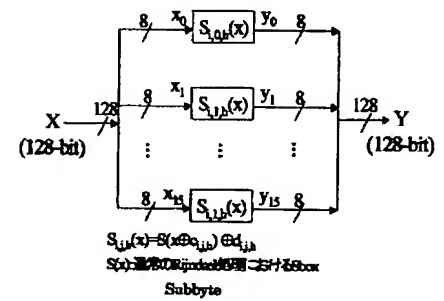
【図21】



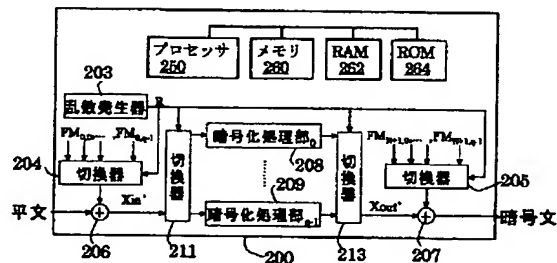
【図26】



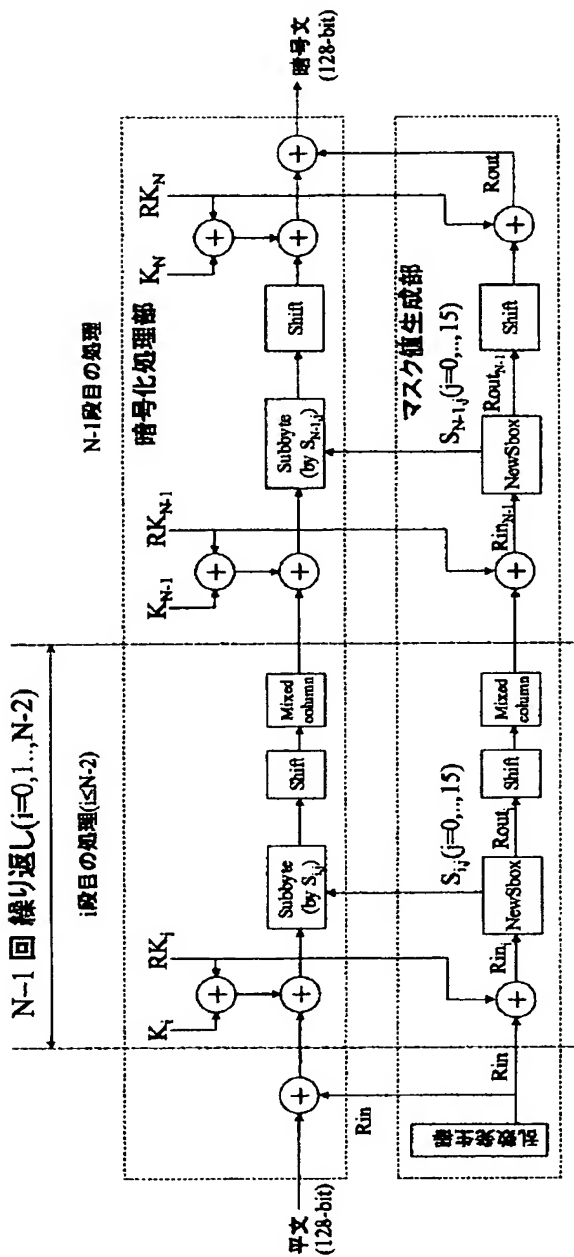
【図28】



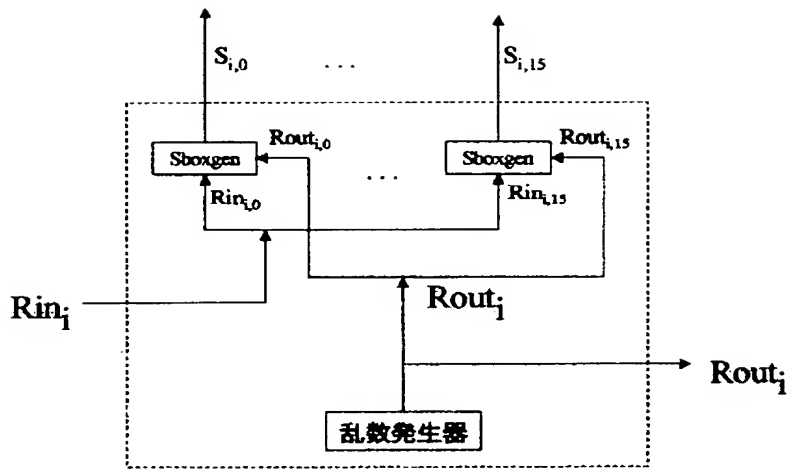
【図24】



【図 19】



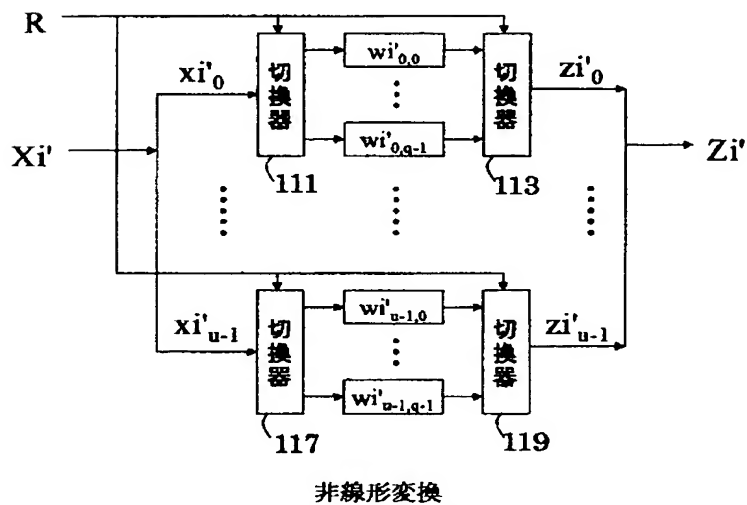
【図20】



Sboxgen:  $S_{i,j}(x) = S(x \oplus Rin_{i,j}) \oplus Rout_{i,j}$  を満たす Sbox,  $S_{i,j}$  の生成

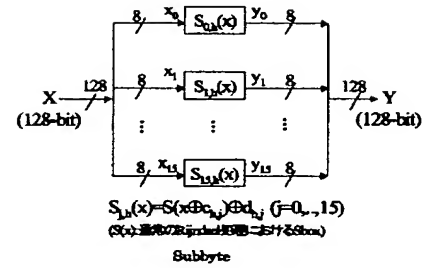
NewSbox

【図23】



非線形変換

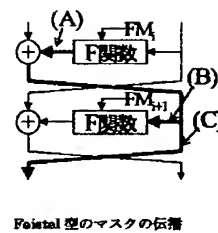
【図30】



$S_{j,i}(x) = S(x \oplus c_{j,i}) \oplus d_{j,i}$  ( $j=0, \dots, 15$ )  
( $c_{j,i}$  と  $d_{j,i}$  は 8-bit 定数)

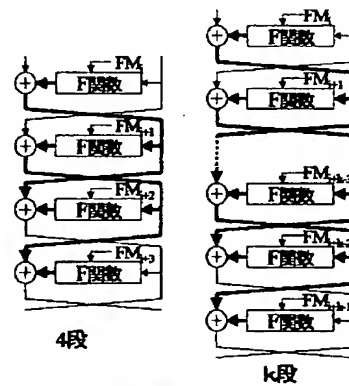
Subbyte

【図35】



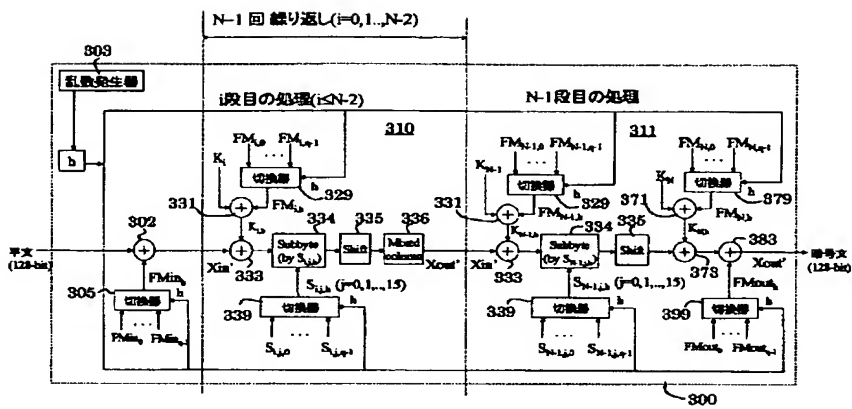
Feistel 型のマスクの伝播

【図36】

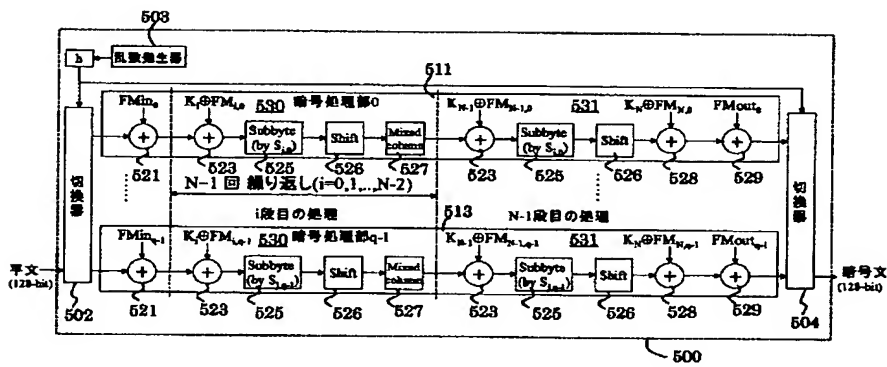


Feistel 型暗号化装置における  
マスク値の発生からキャンセルまでの経路

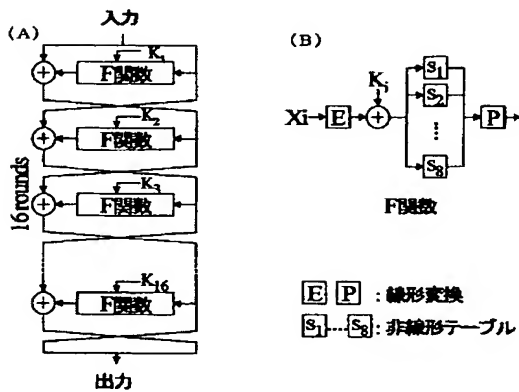
【図 27】



【図 31】

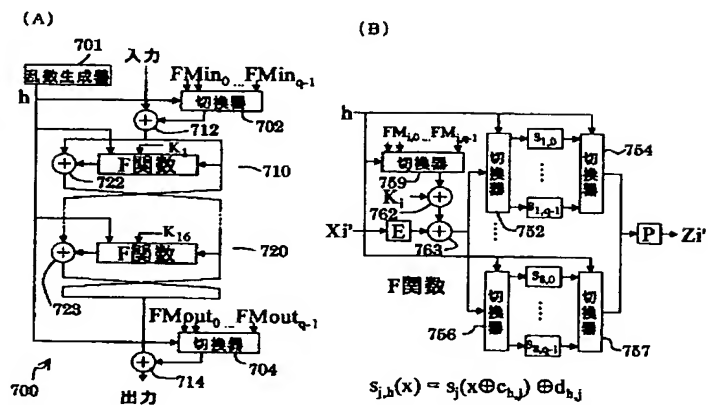


【図 32】



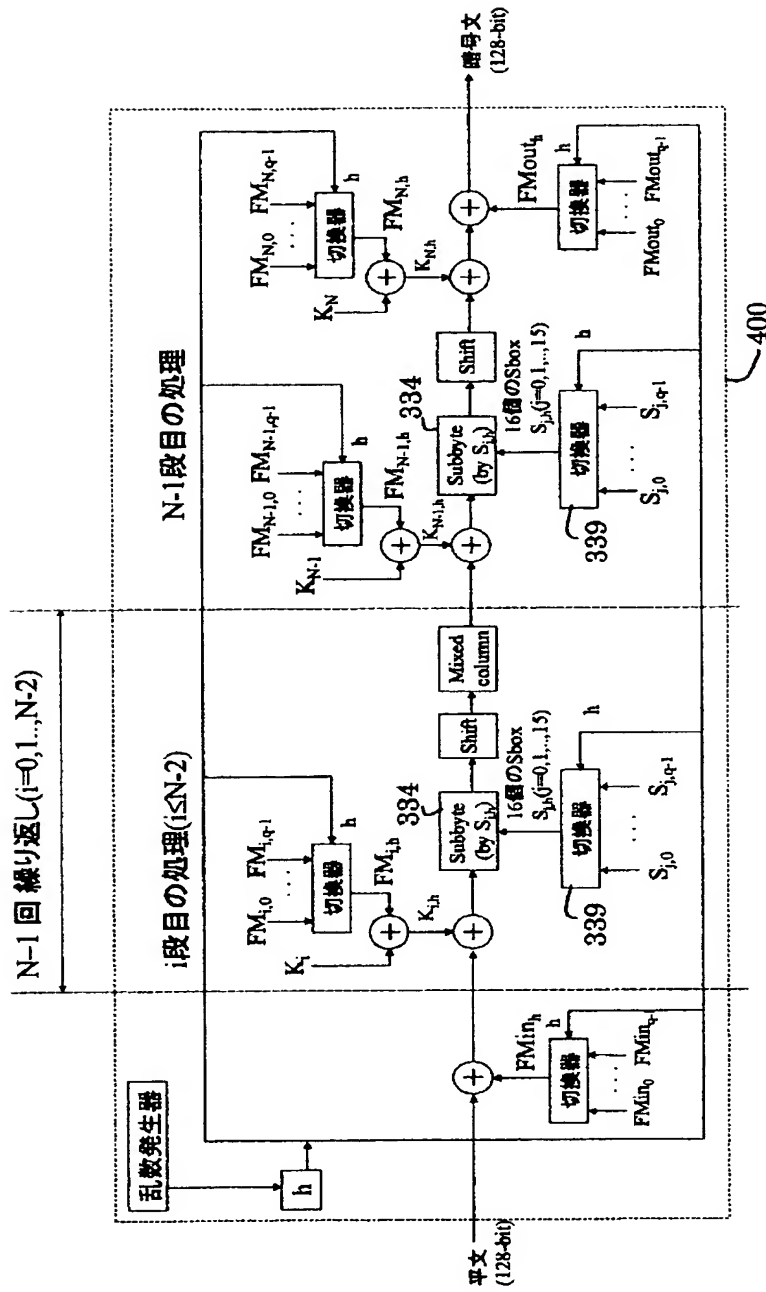
DES の構成

【図 33】

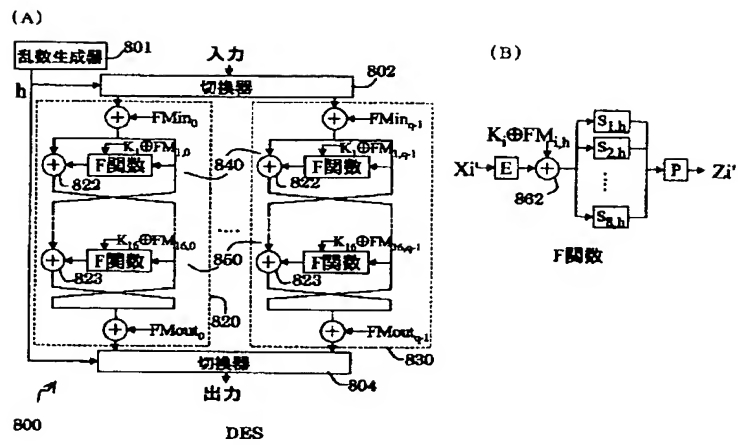


DES

【図29】



【図 34】



フロントページの続き

(72) 発明者 鳥居 直哉  
神奈川県川崎市中原区上小田中 4 丁目 1 番  
1 号 富士通株式会社内

F ターム (参考) 5J104 AA41 AA47 FA00 JA03 NA02